



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - KS 141501**

# **ANALISIS FAKTOR YANG MEMPENGARUHI TINGKAT PERUBAHAN NILAI EFFORT RATE (ER) PADA METODE UCP UNTUK ESTIMASI EFFORT PENGEMBANGAN PERANGKAT LUNAK**

**SON NAVY RAMADHAN KURNIA AL AKBAR**  
**NRP 5209 100 029**

**Dosen Pembimbing I**  
**Sholiq, S.T, M.Kom, M.SA**

**Dosen Pembimbing II**  
**Amna Shifia Nisafani, S.Kom, M.Sc**

**JURUSAN SISTEM INFORMASI**  
**Fakultas Teknologi Informasi**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 2016**



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT - KS 141501**

# **ANALYSIS OF FACTORS AFFECTING THE EFFORT RATE (ER) VALUE CHANGES IN UCP METHOD FOR EFFORT ESTIMATION OF SOFTWARE DEVELOPMENT**

**SON NAVY RAMADHAN KURNIA AL AKBAR  
NRP 5209 100 029**

**Supervisor I  
Sholih, S.T, M.Kom, M.SA**

**Supervisor II  
Amna Shifia Nisafani, S.Kom, M.Sc**

**DEPARTMENT OF INFORMATION SYSTEM  
Faculty of Information Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2016**



## LEMBAR PENGESAHAN

### ANALISIS FAKTOR YANG MEMPENGARUHI TINGKAT PERUBAHAN NILAI EFFORT RATE (ER) PADA METODE UCP UNTUK ESTIMASI EFFORT PENGEMBANGAN PERANGKAT LUNAK

#### TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada

Jurusan Sistem Informasi  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh :

**SON NAVY RAMADHAN KURNIA AL AKBAR**  
**NRP 5209 100 029**

Surabaya, Juli 2016

KETUA  
JURUSAN SISTEM INFORMASI



Dr. Ir. Aris Triyanto, M.Kom.  
**NIP. 19650310 199102 1 001**



## LEMBAR PERSETUJUAN

### ANALISIS FAKTOR YANG MEMPENGARUHI TINGKAT PERUBAHAN NILAI EFFORT RATE (ER) PADA METODE UCP UNTUK ESTIMASI EFFORT PENGEMBANGAN PERANGKAT LUNAK

#### TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Jurusan Sistem Informasi  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh :

**SON NAVY RAMADHAN KURNIA AL AKBAR**  
**NRP 5209 100 029**

Disetujui Tim Penguji : Tanggal Ujian : ..... Juli 2016  
Periode Wisuda : September 2016

**Sholiq, S.T, M. Kom, M.SA**

  
(Pembimbing I)

**Amna Shifia Nisafani, S.Kom, M.Sc**

  
(Pembimbing II)

**Apol Pribadi Subriadi, S.T, M.T**

  
(Penguji I)

**T. Eko Wahyu Tyas D., S. Kom., MBA**

  
(Penguji II)

# **ANALISIS FAKTOR YANG MEMPENGARUHI TINGKAT PERUBAHAN NILAI EFFORT RATE (ER) PADA METODE UCP UNTUK ESTIMASI EFFORT PENGEMBANGAN PERANGKAT LUNAK**

**Nama Mahasiswa** : Son Navy Ramadhan  
**NRP** : 5209 100 029  
**Jurusan** : Sistem Informasi FTIF-ITS  
**Dosen Pembimbing** : Sholiq, S.T., M.Kom., M.SA.  
Amna Shifia Nisafani, S.Kom, M.Sc

## **ABSTRAK**

*Nilai Effort Rate (ER) adalah nilai yang dijadikan variabel untuk menghitung estimasi effort proyek pengembangan perangkat lunak pada metode use case point dengan satuan person-hours. Permasalahan yang timbul yaitu, belum adanya ketetapan berapakah nilai default dari ER yang sesuai untuk digunakan pada perhitungan estimasi effort, serta seringnya perubahan nilai ER yang terjadi lebih dari sekali. Dari beberapa penelitian yang terdahulu juga dapat disimpulkan bahwa terjadi ketidakkonsistenan dalam penentuan nilai ER. Maka, perlu dilakukan analisa faktor-faktor yang berpengaruh terhadap kejadian tersebut agar dapat diketahui penyebab perubahan dan besar pengaruhnya terhadap nilai ER.*

*Metode yang digunakan antara lain: membuat dan menyebarkan kuesioner ke beberapa tim pengembang dengan kondisi satu responden mewakili satu proyek, terakhir melakukan analisa menggunakan teknik analisis dengan Structural Equation Model (SEM) yaitu generalized structured component analysis (GSCA) untuk menguji hipotesis faktor yang mempengaruhi tingkat perubahan nilai ER.*

*Hasil yang didapatkan dari tugas akhir ini yaitu, sumber yang berasal dari internet berpengaruh positif signifikan, komponen perangkat lunak berpengaruh negatif signifikan, teknologi pengembangan berpengaruh negatif tidak signifikan dan metode pengembangan berpengaruh positif tidak signifikan terhadap perubahan nilai ER.*

***Kata kunci: Use Case Point (UCP), Effort Rate (ER), Estimasi Effort, Structural Equation Model, Generalized Structured Component Analysis***

# **ANALYSIS OF FACTORS AFFECTING THE EFFORT RATE (ER) VALUE CHANGES IN UCP METHOD FOR EFFORT ESTIMATION OF SOFTWARE DEVELOPMENT**

**Student Name** : Son Navy Ramadhan  
**NRP** : 5209 100 029  
**Department** : Information Systems FTIF-ITS  
**Supervisor** : Sholiq, S.T., M.Kom., M.SA.  
Amna Shifia Nisafani, S.Kom, M.Sc

## **ABSTRACT**

*Effort Rate (ER) value is the value of the variable used to calculate the estimated effort software development projects on the use case point method with units of person-hours. The problem that arise is, what is the default value of ER is appropriate to be used in the calculation of estimated effort, and the change frequency in ER value that occur more than once. From some previous research can also be inferred that inconsistency occurred in the determination of the ER value. Thus, the need for analysis of the influential factors of the incident in order toknow the cause of the change and its effect on the ER value.*

*The methods used include: create and distribute questionnairse to several teams of developers with the condition one respondent represent one project, lastly, do the analysis using the analysis technique with SEM which is GSCA to test the hypothesis factors that affect the rate of change of the ER value.*

*Results obtained from this research are, sources from the internet has significant positive effect, the software components has significantly negative influence, technology development has insignificant negative effect and development methods has insignificant positive effect against the change in the value of ER.*

***Keyword : Use Case Point (UCP), Effort Rate (ER), Estimated Effort, Structural Equation Model, Generalized Structured Component Analysis***



## DAFTAR ISI

ABSTRAK.....	v
ABSTRACT.....	vii
KATA PENGANTAR .....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL.....	xv
1 BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah .....	1
1.2 Perumusan Masalah .....	4
1.3 Batasan Masalah .....	4
1.4 Tujuan Penelitian .....	4
1.5 Manfaat Penelitian .....	5
1.6 Sistematika Penulisan .....	5
1.7 Relevansi.....	6
BAB II TINJAUAN PUSTAKA .....	9
2.1 Studi Sebelumnya .....	9
2.1.1 Penelitian Terkait Mengenai <i>Effort Rate (ER)</i> .....	9
2.1.2 Faktor Dugaan Penyebab Perubahan .....	12
2.2 Dasar Teori .....	12
2.2.1 Metode Use Case Point .....	12
2.2.2 Estimasi Effort.....	14
2.2.3 SPSS .....	14
2.2.4 Structural Equational Modeling (SEM).....	17
BAB III METODOLOGI PENELITIAN .....	23
3.1 Studi Pustaka.....	24
3.2 Penggalan Kerangka Konseptual .....	24
3.3 Pembentukan Hipotesa.....	24
3.4 Peninjauan Responden .....	24
3.5 Pembuatan Kuesioner .....	25
3.6 Uji Validitas dan Uji Reliabilitas .....	25
3.7 Analisis Inferensial .....	25

3.8 Uji Hipotesis .....	26
3.9 Kesimpulan .....	26
<b>BAB IV PERANCANGAN .....</b>	<b>27</b>
4.1 Penggalian Kerangka Konseptual .....	27
4.2 Pembentukan Hipotesa.....	30
4.3 Pembuatan Kuesioner .....	33
4.3.1 Prosedur Analisa.....	33
4.3.2 Instrumen Penelitian.....	42
4.4 Peninjauan Responden .....	52
<b>BAB V IMPLEMENTASI.....</b>	<b>53</b>
5.1 Penyebaran Kuesioner .....	53
5.2 Pengolahan Data. ....	54
5.3 Hambatan Implementasi .....	55
5.4 Solusi Hambatan .....	55
<b>BAB VI HASIL DAN PEMBAHASAN .....</b>	<b>57</b>
6.1 Uji Asumsi Kualitas Pengukuran.....	57
6.1.1 Uji validitas .....	57
6.1.2 Uji Reliabilitas.....	62
6.1.3 Uji Linieritas.....	62
6.2 Analisis Inferensial .....	64
6.2.1 Identifikasi Goodness of FIT.....	69
6.2.2 Identifikasi R Square .....	71
6.2.3 Identifikasi Indikator-Indikator .....	72
6.3 Pengujian Hipotesis .....	78
6.4 Interpretasi Hasil Uji Hipotesis.....	83
<b>BAB VII KESIMPULAN DAN SARAN .....</b>	<b>87</b>
7.1 Kesimpulan .....	87
7.2 Saran .....	88
<b>DAFTAR PUSTAKA .....</b>	<b>91</b>
<b>LAMPIRAN A.....</b>	<b>A - 1</b>
<b>LAMPIRAN B .....</b>	<b>B - 29</b>
<b>LAMPIRAN C .....</b>	<b>C - 37</b>

## **DAFTAR GAMBAR**

Gambar 1.1 Roadmap Laboratorium Manajemen Sistem Informasi (MSI) .....	7
Gambar 3.1 Metodologi Pengerjaan Tugas Akhir .....	23
Gambar 4.1 Konseptual Model .....	28
Gambar 4.2 Acuan Hipotesis .....	31
Gambar 6.1 Hasil Uji Hipotesis .....	80

*Halaman ini sengaja dikosongkan*



## DAFTAR TABEL

Tabel 2.1 Perbandingan antara SEM berbasis variance-GSCA dengan sem berbasis covariance .....	20
Tabel 4.1 Technical Factor dan bobotnya.....	34
Tabel 4.2 Environmental Factor dan bobotnya .....	35
Tabel 4.3 Indikator Metode Pengembangan Perangkat Lunak .....	35
Tabel 4.4 Indikator Teknologi Pengembangan Perangkat Lunak .....	36
Tabel 4.5 Indikator Komponen Perangkat Lunak .....	37
Tabel 4.6 Sources dari Internet .....	38
Tabel 4.7 The Soft Factors.....	39
Tabel 4.8 Productivity Factor.....	40
Tabel 4.9 Pertanyaan Tentang Metode Pengembangan Perangkat Lunak .....	43
Tabel 4.10 Pertanyaan Tentang Teknologi Pengembangan Perangkat Lunak .....	44
Tabel 4.11 Pertanyaan Tentang Komponen Perangkat Lunak .....	46
Tabel 4.12 Pertanyaan Tentang Sources dari Internet .....	48
Tabel 4.13 Pertanyaan Tentang Productivity Factor.....	48
Tabel 5.1 Daftar Koresponden.....	53
Tabel 6.1 Uji Validitas MPPL .....	58
Tabel 6.2 Uji Validitas TPPL.....	58
Tabel 6.3 Uji Validitas KPL .....	59
Tabel 6.4 Uji Validitas SDI .....	60
Tabel 6.5 Uji Validitas Productivity Factor .....	60
Tabel 6.6 Uji Reliabilitas Variabel Laten .....	62
Tabel 6.7 Hasil Uji Linieritas.....	63
Tabel 6.8 Goodness of Fit dari Model Keseluruhan .....	68
Tabel 6.9 Kriteria SRMR .....	68
Tabel 6.10 Model Fit.....	69
Tabel 6.11 Identifikasi R Square.....	71

Tabel 6.12 Identifikasi Indikator MPPL .....	72
Tabel 6.13 Identifikasi Indikator TPPL .....	73
Tabel 6.14 Identifikasi Indikator KPL .....	75
Tabel 6.15 Identifikasi Indikator SDI .....	76
Tabel 6.16 Identifikasi Indikator PF .....	77
Tabel 6.17 Path Coefficients .....	79
Tabel 6.18 Hasil Hipotesis .....	82

# **BAB I**

## **PENDAHULUAN**

Pada bab ini, akan dijelaskan tentang Latar Belakang Masalah, Perumusan Masalah, Batasan Masalah, Tujuan Tugas Akhir, dan Relevansi atau Manfaat Kegiatan Tugas Akhir.

### **1.1 Latar Belakang Masalah**

Dalam proyek pembuatan maupun pengembangan perangkat lunak, salah satu tahap pentingnya adalah tahapan planning atau perencanaan. Dalam tahapan planning, upaya dan estimasi sumber daya harus sangat diperhitungkan, dengan mempertimbangkan manusia (*human resource*), material, infrastruktur, waktu, dan anggaran [1].

Estimasi effort terdiri dari kata estimasi dan effort. Estimasi sendiri adalah sebuah pengukuran atau penilaian yang didasarkan pada hasil secara kuantitatif atau tingkat akurasi dapat diukur dengan angka [2]. Sedangkan, Definisi *effort* adalah suatu pekerjaan secara real yang kita lakukan dalam menyelesaikan suatu proyek, dengan satuan *staff-days* atau *staff-hours* [3]. Dilihat dari pengertian yang telah dijelaskan diatas, maka dapat disimpulkan pengertian dari estimasi *effort* adalah suatu kegiatan yang melakukan prediksi atau ramalan mengenai berapa banyak jumlah pekerja dan berapa lama waktu yang diperlukan untuk menyelesaikan sebuah proyek, di sini proyek yang dimaksud berkaitan dengan pembuatan atau pengembangan perangkat lunak.

Dewasa ini metode *Use Case Point* (UCP) menjadi hal yang menarik dalam estimasi *effort* proyek perangkat lunak. Berdasarkan hasil penelitian, ditunjukkan bahwa metode UCP

dapat berhasil digunakan untuk memperkirakan *effort* pengembangan perangkat lunak [4]. Metode UCP ini sangat penting untuk melakukan estimasi *effort*, dimana nilai dari estimasi *effort* merupakan hasil dari nilai UCP dikalikan dengan nilai *Effort Rate* (ER). Nilai *Effort Rate* (ER) sendiri adalah nilai yang dijadikan variabel untuk menghitung *effort*.

*Effort Rate* (ER) adalah nilai yang pertama kali dipopulerkan oleh Karner pada tahun 1993, yang pada saat itu memiliki nilai 20 *staff-hours* [5]. Nilai *Effort Rate* tersebut didapatkan dari tiga data proyek pengembangan perangkat lunak dengan menggunakan analisis regresi. Dasar analisa dari tiga data yang diperoleh Karner tersebut masih bisa diperdebatkan lagi, karena Karner tidak menggunakan analisis korelasi antar data untuk membentuk persamaan regresi, sedangkan analisis regresi terhadap tiga data diskrit tersebut cenderung tidak akurat [5]. Sebelum adanya usulan dari para ahli lainnya, dalam pengembangan penelitian terdahulu masih menggunakan acuan nilai *Effort Rate* yang diusulkan oleh Karner. Kemudian di tahun 1998, Schneider & Winters mengusulkan nilai *Effort Rate* sebesar 20, 28, dan 36 *staff-hours* dengan menggunakan dasar kompleksitas proyek yang mengacu pada *Technical Complexity Factor* (TCF) [6]. Di tahun 2006, Clemmons mengusulkan nilai *Effort Rate* sebesar 18 *staff-hours* berdasarkan kualitas tim proyek dan data historis [7]. Dan lima tahun kemudian (2011), nilai *Effort Rate* mengalami perkembangan yang berkisar antara 4 sampai 35 *staff-hours* yang dipopulerkan oleh Ochodek berdasarkan dari perhitungan 14 proyek perangkat lunak yang telah selesai [8]. Temuan terkini di tahun 2013, Subriadi et.al mengusulkan nilai *Effort Rate* sebesar 8.2 *staff-hours* dari penelitian menggunakan metode *Use Case Point* (UCP) [9].



Setelah melalui berbagai penelitian terhadap perhitungan nilai *Effort Rate*, maka ditemukan adanya permasalahan, yakni belum adanya ketentuan berapakah nilai default dari *Effort Rate*, serta seringnya perubahan nilai *Effort Rate* yang terjadi lebih dari sekali. Padahal, nilai ER sangat dibutuhkan dalam perhitungan estimasi *effort* yaitu, nilai ER dikali dengan nilai UCP [5]. Bila nilai ER sering berubah-ubah, maka akan sangat sulit untuk melakukan estimasi *effort* pada suatu proyek pengembangan perangkat lunak. Pada penelitian sebelumnya, hasil analisa nilai ER membentuk beberapa dugaan mengapa nilai *Effort Rate* berubah-ubah. Adapun dugaannya ialah, semakin majunya teknologi *software engineering*, adanya metode *software engineering*, adanya *software by component*, adanya *source* dari *internet* [9].

Penelitian tugas akhir ini melalui beberapa tahapan meliputi, pembuatan kuesioner untuk keperluan pengumpulan data analisis, setelah dibuat, kuesioner disebarkan ke beberapa responden yang termasuk dalam anggota tim pengembang dan satu responden mewakili satu proyek, lalu, data hasil kuesioner diuji validitas dan reliabilitasnya, terakhir melakukan analisa korelasi dan regresi linear berganda antara data hasil kuesioner dengan data nilai ER. Beberapa tahapan tersebut berfungsi untuk mencari hubungan dari masing-masing faktor dan juga pengaruhnya pada perubahan nilai *Effort Rate*.

## 1.2 Perumusan Masalah

Permasalahan yang dihadapi dalam penelitian ini antara lain adalah sebagai berikut :

1. Faktor – faktor apa saja yang mempengaruhi tingkat perubahan nilai *Effort Rate* (ER)?
2. Apakah ada hubungan positif dari masing – masing faktor dengan tingkat perubahan nilai *Effort Rate* (ER)?
3. Berapa saja pengaruh masing – masing faktor terhadap tingkat perubahan nilai *Effort Rate* (ER)?

## 1.3 Batasan Masalah

Batasan masalah pada penelitian ini antara lain:

- 1 Data yang digunakan untuk melakukan analisa didapatkan dari responden, dimana responden adalah orang yang termasuk dalam anggota tim pengembang perangkat lunak yang berbasis *Object Oriented Project* (OOP) dan satu responden mewakili satu proyek.
- 2 Data nilai ER didapatkan dari hasil penelitian terdahulu dengan menggunakan metode UCP pada proyek perangkat lunak.
- 3 Jenis proyek pembuatan atau pengembangan perangkat lunak yang dijadikan penelitian tidak dilihat dari sisi ukurannya (besar, sedang, atau kecil).

## 1.4 Tujuan Penelitian

Tujuan penelitian ini yaitu,

- 1 Mengetahui faktor yang mempengaruhi perubahan nilai *Effort Rate* (ER).
- 2 Mengetahui hubungan serta pengaruh masing – masing faktor terhadap tingkat perubahan nilai *Effort Rate* (ER).

## **1.5 Manfaat Penelitian**

Manfaat penulisan penelitian ini adalah diketahuinya faktor – faktor yang berpengaruh pada tingkat perubahan nilai *Effort Rate* (ER) pada proyek pembuatan dan pengembangan perangkat lunak bagi dunia akademik dan faktor – faktor tersebut dapat dijadikan pertimbangan pada saat proyek pembuatan atau pengembangan perangkat lunak akan dimulai bagi pengembang perangkat lunak.

## **1.6 Sistematika Penulisan**

Sistematika penulisan buku tugas akhir ini dibagi menjadi enam bab sebagai berikut:

### **BAB I PENDAHULUAN**

Pada bab ini menjelaskan tentang latar belakang masalah, perumusan masalah, batasan masalah, tujuan, manfaat penelitian , target luaran dan sistematika penulisan.

### **BAB II TINJAUAN PUSTAKA**

Pada bab ini akan diuraikan istilah-istilah yang digunakan pada penulisan buku tugas akhir ini serta dasar teori yang digunakan pada penelitian ini.

### **BAB III METODOLOGI PENELITIAN**

Pada bab ini akan dijelaskan mengenai metode yang digunakan dalam penulisan tugas akhir. Metode ini digunakan sebagai panduan dalam pengerjaan penelitian agar berjalan terarah dan sistematis. Metode yang digunakan terangkum dalam sebuah diagram alur yang sistematis dan dijelaskan tahap demi tahap.

### **BAB IV PERANCANGAN**

Pada bab ini berisi rancangan bagaimana penelitian akan dilakukan, pemilihan obyek penelitian, dan strategi pelaksanaan, dari konsep hingga kuesioner terbentuk.

## **BAB V IMPLEMENTASI**

Pada bab ini berisi bagaimana penelitian dilakukan, persebaran kuesioner, hambatan dan rintangan dalam pelaksanaan, tanggapan responden hingga data yang dibutuhkan terpenuhi.

## **BAB VI HASIL DAN PEMBAHASAN**

Pada bab ini akan dilakukan penjabaran hasil dan pembahasan terhadap hasil yang diperoleh digunakan untuk menjawab rumusan masalah yang diangkat dalam penelitian ini.

## **BAB VI KESIMPULAN DAN SARAN**

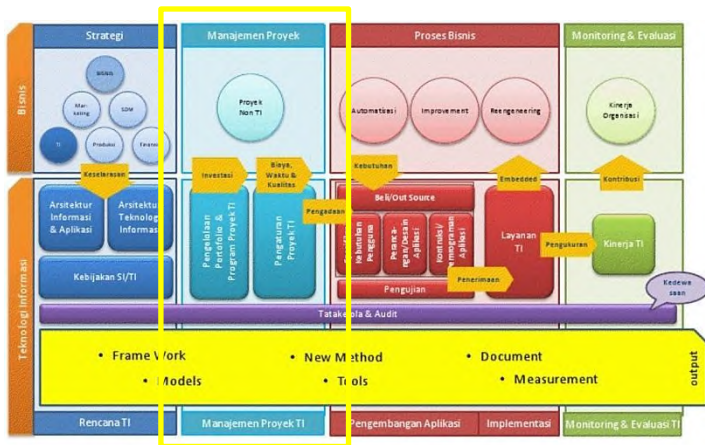
Pada Bab ini berisi kesimpulan dari keseluruhan permasalahan penelitian dan saran perbaikan yang dapat dikembangkan di masa mendatang untuk penelitian ini.

### **1.7 Relevansi**

Relevansi pengerjaan tugas akhir ini yaitu terkait dengan manajemen proyek. Manajemen proyek terdiri dari pengelolaan portofolio, program, dan pengaturan proyek. Gambar 1.1 adalah roadmap penelitian dari Laboraturium Manajemen Sistem informasi (PPSI) ITS.

Pada gambar 1.1 menunjukan bahwa biaya dan waktu merupakan elemen pembentuk dari pengaturan proyek TI yang merupakan kelanjutan dari Manajemen Proyek Teknologi Informasi (TI). Output yang ada dalam pengaturan proyek TI ini yaitu berupa dokumentasi pengatuan proyek.





**Gambar 1.1 Roadmap Laboratorium Manajemen Sistem Informasi (MSI)**

*Halaman ini sengaja dikosongkan*

## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bagian ini akan dipaparkan tentang konsep apa saja yang akan digunakan atau diterapkan dalam menyelesaikan penelitian ini. Dengan adanya tinjauan pustaka ini diharapkan dapat memberikan penjelasan dan gambaran secara umum mengenai penelitian ini. Adapun konsep yang akan digunakan pada tugas akhir ini adalah sebagai berikut.

#### **2.1 Studi Sebelumnya**

##### **2.1.1 Penelitian Terkait Mengenai *Effort Rate (ER)***

Nilai *Effort Rate (ER)* adalah nilai yang dijadikan variabel untuk menghitung *effort* dengan satuan *person-hours* [3]. Nilai *Effort Rate (ER)* didapatkan setelah dilakukan pembagian antara nilai aktual *effort* dengan nilai *use case point (UCP)*. Berdasarkan beberapa penelitian yang telah dilakukan di Eropa dan Amerika didapatkan nilai *Effort Rate (ER)* sebagai berikut:

1. Nilai *Effort Rate (ER)* *use case point* sebesar 20 [10]. Pada tahun 1993, Karner melakukan penelitian menggunakan 3 project. Hasil perhitungan yang didapatkan yaitu 3 nilai *Effort Rate (ER)*, masing-masing 20 ER; 20,9 ER dan 28, 7 ER. Setelah didapatkan nilai *Effort Rate (ER)* dari ke tiga project tersebut maka perlu dilakukan penarikan garis linier, sehingga mendapatkan nilai empiris *Effort Rate (ER)* sebesar 20. Penelitian yang dilakukan oleh Karner ini mengasumsikan semua faktor dalam TCF berbobot 3, sehingga nilai TCF yang dihasilkan sudah pasti bernilai 1. Hal ini tidak sesuai apabila diterapkan pada proyek pengembangan perangkat lunak yang lain, sebab semua faktor dalam TCF belum tentu mempunyai bobot sebesar 3 (bisa berbobot 0 sampai 5). Dengan bobot

yang berbeda-beda tersebut tentunya akan menghasilkan nilai TCF yang berbeda pula. Maka dari itu penelitian dari Karner ini masih belum bisa di uji keempirisannya

2. Nilai *Effort Rate* (ER) *Use case point* sebesar 20, 24, dan 36 [6].

Nilai *default* untuk *Effort Rate* (ER) diusulkan oleh Karner adalah 20 *hour* per UCP. Schneider dan kelompoknya mengusulkan sebuah metode untuk menentukan nilai awal *Effort Rate* (ER). Berdasarkan pengalaman, mereka menyarankan untuk menghitung jumlah *Enviromental Factors* (EF) pada faktor F1-F6 yang mempengaruhi diperkirakan kurang dari 3 dan faktor F7-F8 yang mempengaruhi diperkirakan akan lebih besar dari 3. Jika total dihitung adalah sama dengan 2 atau kurang, maka nilai *Effort Rate* (ER) yaitu 20 *hour* per UCP harus digunakan. Jika total adalah antara 3 dan 4, mereka menyarankan menggunakan *Effort Rate* (ER) sama dengan 28 *hour* per UCP. Jika jumlah yang dihitung lebih besar dari 4, maka nilai *Effort Rate* (ER) adalah 36 *hours* per UCP harus digunakan. (dalam hal ini proyek tersebut dianggap sebagai salah satu yang sangat berisiko). Namun, nilai dari *Enviromental Factors* (EF) pada faktor F1-F6 belum tentu bisa dipastikan kurang dari 3 dan faktor F7-F8 belum tentu bisa dipastikan akan lebih besar dari 3. Maka dari itu, nilai *Effort Rate* (ER) yang diusulkan oleh Schneider ini kurang sesuai apabila digunakan pada proyek pengembangan perangkat lunak yang mempunyai *Enviromental Factors* (EF) pada

faktor F1-F6 lebih dari 3, dan faktor F7-F8 kurang dari 3.

3. Nilai *Effort Rate* (ER) *use case point* sebesar 18 [7].  
 Pada penelitian yang dilakukan Clemmons pada tahun 2006, didapatkan nilai UCP sebesar 120. Nilai *Effort Rate* (ER) tersebut didapatkan setelah melakukan pembagian antara nilai aktual *Effort Rate* (ER) sebesar 2200 dengan *use case point* estimasi sebesar 120.  

$$\text{Effort Rate (ER)} = 2200 / 120 = 18 \text{ person-hours}$$
4. Nilai *Effort Rate* (ER) *use case point* bervariasi antara 4 sampai 35 [8].  
 Pada tahun 2011 tim penelitian yang diketuai oleh Ochodek melakukan perhitungan *effort* dengan meneliti 14 studi kasus dengan nilai aktual *effort* berkisar antara 277 sampai 3593 *person-hours*. Dalam penelitiannya, disebutkan bahwa “menggunakan nilai default untuk *Effort Rate* (ER) adalah suatu keharusan jika sebuah organisasi tidak memiliki data historis mengenai produktivitas” [8].
5. Nilai *Effort Rate* (ER) *use case point* yang digunakan oleh Kusumoto mengacu pada karnier, yaitu 20 [11].  
 Pada penelitian yang dilakukan oleh Kusumoto dan kelompoknya menjelaskan pengalaman aktual memperkenalkan metode *Use case point* (UCP) untuk proyek perangkat lunak di perusahaan. Untuk pengenalan yang efektif dari metode UCP, mereka mengembangkan *tool* pengukur *use case* otomatis yang disebut U-EST. Mereka juga membandingkan nilai UCP yang dihitung oleh alat dengan spesialis. Hasilnya, UCP diukur dengan U-EST mirip dengan yang dihitung oleh spesialis.

*Effort* dihitung dengan mengalikan nilai spesifik (*person-hours*) dan UCP. Saat ini, nilai yang digunakan oleh Kusumoto dan kelompoknya yaitu *20 hours per UCP* seperti yang diusulkan oleh Karner. Tapi, nilai tersebut dapat diubah melalui GUI [11].

6. Nilai *Effort Rate* (ER) *use case point* yang digunakan oleh Edward R Carroll mengacu pada Schneider and Winters, yakni menggunakan kisaran 20 dan 28 [12].
7. Nilai *Effort Rate* (ER) *use case point* yang digunakan oleh Yavari mengacu pada Schneider and Winters, yakni menggunakan kisaran 20 dan 28 [13].
8. Nilai *Effort Rate* (ER) *use case point* yang diusulkan Subriadi et.al setelah melalui penelitian proyek pengembangan perangkat lunak di bidang bisnis dengan menggunakan metode *use case point* pada tahun 2013 adalah sebesar *8.2 staff-hours* [9].

## **2.1.2 Faktor Dugaan Penyebab Perubahan**

Hasil analisis nilai *Effort Rate* (ER) pada penelitian yang lalu mengungkapkan beberapa dugaan alasan mengapa nilai ER yang didapatkan lebih kecil daripada yang dikemukakan oleh Karner [9]. Adapun dugaan alasan – alasan tersebut adalah, Teknologi Software Engineering, Metode Software Engineering, Software by Component, Software By Framework, tersedia Source dari Internet [9].

## **2.2 Dasar Teori**

### **2.2.1 Metode Use Case Point**

Gustav Karner mengembangkan metode *Use Case Point* (UCP) yang merupakan turunan atau adaptasi dari metode *Function Point Analysis* (FPA) yang bertujuan untuk

menyediakan metode estimasi sederhana yang disesuaikan dengan orientasi pada objek proyek perangkat lunak [7].

Metode *Use Case Point (UCP)* sebenarnya bukan metode baru, tetapi belum menjadi populer meskipun mudah untuk dipelajari. Namun, akhir-akhir ini metode *UCP* menjadi topik menarik dalam estimasi proyek perangkat lunak. Hasil penelitian menunjukkan bahwa metode *UCP* berhasil digunakan untuk memperkirakan *effort* pengembangan perangkat lunak [4]. Selain itu, metode ini memberikan estimasi yang hampir mendekati estimasi sebenarnya yang dihasilkan dari pengalaman pembuatan atau pengembangan perangkat lunak. Hal tersebut dibuktikan oleh beberapa penelitian yang pernah dilakukan sebelumnya, dan menghasilkan pernyataan sebagai berikut:

1. S. Nageswaran dalam "Test Effort Estimation" menjelaskan, UCP memiliki deviasi sebesar 6% atau tingkat akurasi mencapai 94% [14].
2. Di "Proceedings of Empirical Assessment in Software Engineering", menurut B. Anda adalah UCP memiliki deviasi sebesar 19% atau tingkat akurasi mencapai 81%, sementara estimasi para ahli memiliki deviasi sebesar 20% atau tingkat akurasi mencapai 80% [4].
3. Pendapat E. R. Carroll saat *OOPSLA Conference* adalah UCP memiliki deviasi sebesar 9% atau tingkat akurasi mencapai 91% [12].

Setelah dilakukan perhitungan menggunakan metode *use case point*, maka akan dihasilkan nilai estimasi *effort* pembuatan atau pengembangan suatu proyek perangkat lunak. Nilai estimasi *effort* tersebut didapatkan setelah melalui perkalian nilai *use case point* estimasi dengan nilai *Effort Rate (ER)*.

Metode *use case* seringkali digunakan untuk mengetahui kebutuhan fungsional suatu proyek software. Pemodelan *use case* merupakan suatu teknik yang telah digunakan sebagian besar industri untuk menggambarkan dan

mengetahui kebutuhan fungsional dari sebuah sistem software.

### 2.2.2 Estimasi Effort

Estimasi yang dilakukan pada penelitian ini diaplikasikan dalam proyek pengembangan perangkat lunak. Definisi dari estimasi perangkat lunak yaitu suatu kegiatan melakukan prediksi atau ramalan mengenai keluaran dari sebuah proyek dengan meninjau jadwal, usaha, biaya bahkan hingga ke resiko yang akan ditanggung dalam proyek tersebut [10]. Meski estimasi tidak mungkin dapat menghasilkan sebuah hasil yang sangat akurat, tetapi ketidakakuratan tersebut dapat diminimalkan dengan menggunakan beberapa metode yang sesuai dengan proyek yang akan dilakukan estimasi.

Penelitian ini mengangkat *estimasi effort* pada proyek pengembangan perangkat lunak. *Effort* adalah kerja real yang kita lakukan dalam menyelesaikan suatu proyek. Satuannya adalah *man-days* atau *man-hour*. Misalnya suatu aplikasi diestimasi membutuhkan *effort* 10 *man days*. Artinya aplikasi ini akan selesai bila dikerjakan 1 orang selama 10 hari terus menerus atau 5 hari bila ada 2 pekerja. *Effort* tidak mempertimbangkan libur ataupun cuti [3].

Dari pengertian estimasi dan *effort* di atas, maka dapat disimpulkan bahwa *Estimasi effort* adalah suatu kegiatan melakukan prediksi atau ramalan mengenai berapa banyak pekerja dan berapa lama waktu yang diperlukan untuk menyelesaikan proyek tersebut. *Estimasi effort* pada penelitian ini akan didapatkan setelah melakukan perhitungan menggunakan metode *use case point* (UCP).

### 2.2.3 SPSS

SPSS adalah sebuah program komputer yang digunakan untuk membuat analisis statistika. SPSS dipublikasikan oleh SPSS Inc. SPSS (Statistical Package for the Social Sciences



atau Paket Statistik untuk Ilmu Sosial) versi pertama dirilis pada tahun 1968, diciptakan oleh Norman Nie, seorang lulusan Fakultas Ilmu Politik dari Stanford University, yang sekarang menjadi Profesor Peneliti Fakultas Ilmu Politik di Stanford dan Profesor Emeritus Ilmu Politik di University of Chicago. SPSS adalah salah satu program yang paling banyak digunakan untuk analisis statistika ilmu sosial. SPSS digunakan oleh peneliti pasar, peneliti kesehatan, perusahaan survei, pemerintah, peneliti pendidikan, organisasi pemasaran, dan sebagainya. Selain analisis statistika, manajemen data (seleksi kasus, penajaman file, pembuatan data turunan) dan dokumentasi data (kamus metadata ikut dimasukkan bersama data) juga merupakan fitur-fitur dari software dasar SPSS [15].

SPSS adalah aplikasi yang digunakan untuk membuat analisis statistik. SPSS dipublikasikan oleh SPSS Inc. yang kemudian diakuisisi oleh IBM. Statistik yang termasuk software dasar SPSS yaitu:

- **Statistik Deskriptif:** Tabulasi Silang, Frekuensi, Deskripsi, Penelusuran, Statistik Deskripsi Rasio.
- **Statistik Bivariat:** Rata-rata, t-test, ANOVA, Korelasi (bivariat, parsial, jarak), Nonparametric tests.
- **Prediksi Hasil Numerik:** Regresi Linear.
- **Prediksi untuk mengidentifikasi kelompok:** Analisis Faktor, Analisis Cluster (two-step, K-means, hierarkis), Diskriminan.

SPSS dapat membaca dan menulis data dari file teks ASCII (termasuk file hierarkis), paket statistik lainnya, spreadsheets dan database. SPSS dapat membaca dan menulis ke dalam tabel database eksternal relasional melalui ODBC dan SQL.

### **2.2.3.1 Uji Validitas**

Uji validitas dilakukan untuk menunjukkan sejauh mana suatu alat pengukur mengukur dengan tepat yang ingin diukur. Suatu skala pengukuran disebut valid bila melakukan apa yang seharusnya diukur [16]. Uji validitas kuesioner dilakukan dengan menggunakan software SPSS dengan nilai korelasi pearson antara satu pertanyaan dengan nilai total dalam satu variabel. Koefisien validitas yang tidak begitu tinggi, katakanlah berada di sekitar angka 0,50 akan lebih dapat diterima dan dianggap memuaskan daripada koefisien reliabilitas dengan angka yang sama [17]. Namun, apabila koefisien validitas itu kurang daripada 0,30 biasanya dianggap sebagai tidak memuaskan.

Sisi lain dari pengertian validitas adalah aspek kecermatan pengukuran. Suatu alat ukur yang valid tidak hanya mampu menghasilkan data yang tepat akan tetapi juga harus memberikan gambaran yang cermat mengenai data tersebut.

### **2.2.3.2 Uji Reliabilitas**

Reliabilitas menunjukkan konsistensi dan stabilitas dari suatu skor (skala pengukuran) [16]. Pada dasarnya uji reliabilitas menunjukkan sejauh mana suatu alat ukur dapat memberikan hasil yang relatif sama bila dilakukan pengukuran kembali pada subyek yang sama.

Nilai koefisien reliabilitas yang baik adalah diatas 0,6 (cukup baik), di atas 0,8 (baik).

Menurut Sukadji [15], reliabilitas suatu tes adalah seberapa besar derajat tes mengukur secara konsisten sasaran yang diukur. Reliabilitas dinyatakan dalam bentuk angka, biasanya sebagai koefisien. Koefisien tinggi berarti reliabilitas tinggi. Menurut Nursalam [15], Reliabilitas adalah kesamaan hasil pengukuran atau pengamatan bila fakta atau kenyataan hidup tadi diukur atau diamati berkali – kali dalam waktu yang berlainan. Alat dan cara mengukur atau

mengamati sama – sama memegang peranan penting dalam waktu yang bersamaan.

Berdasarkan beberapa pendapat tentang pengertian reliabilitas diatas, maka dapat diambil kesimpulan bahwa reliabilitas adalah suatu keabsahan suatu tes untuk mengukur atau mengamati sesuatu yang menjadi objek ukur.

### **2.2.3.3 Uji Liniertitas**

Uji linearitas bertujuan untuk menganalisis besarnya pengaruh variabel bebas terhadap variabel terikat. Selain itu, metode ini juga dapat digunakan untuk menentukan bentuk hubungan antara kedua variabel sekaligus korelasi di antara keduanya [16]. Data yang memiliki hubungan kuat akan tampak berdekatan. Sedangkan data yang memiliki hubungan yang lemah akan tampak tersebar.

Uji linearitas dilakukan dengan mencari persamaan garis regresi variabel bebas X terhadap variabel terikat Y. Untuk ketentuannya, yaitu jika F hitung (regresi) lebih besar dari harga F tabel pada taraf signifikansi 5% ( $\alpha = 0,05$ ), maka harga F hitung (regresi) signifikan, yang berarti bahwa koefisien regresi adalah berarti (bermakna).

## **2.2.4 Structural Equational Modeling (SEM)**

### **2.2.4.1 Definisi SEM**

Structural Equation Modeling (SEM) adalah teknik analisis statistika yang mengkombinasikan beberapa aspek yang terdapat pada analisis jalur dan analisis faktor konfirmatori untuk mengestimasi beberapa persamaan secara simultan. Model persamaan struktural merupakan generasi kedua teknik analisis multivariat yang memungkinkan peneliti untuk menguji hubungan antara variabel yang kompleks untuk memperoleh gambaran menyeluruh mengenai keseluruhan model. Dengan kata lain, Structural Equation Modeling (SEM) merupakan teknik statistik yang digunakan untuk

membangun dan menguji model statistik yang biasanya dalam bentuk model-model sebab akibat. SEM sebenarnya merupakan teknik hibrida yang meliputi aspek-aspek penegasan (confirmatory) dari analisis faktor, analisis jalur dan regresi yang dapat dianggap sebagai kasus khusus dalam SEM.

Dari definisi diatas dapat disimpulkan bahwa SEM mempunyai karakteristik yang bersifat sebagai teknik analisis untuk lebih menegaskan (confirm) daripada untuk menerangkan.

Menurut Kaplan dalam [17], Structural Equation Modeling (SEM) merupakan perpaduan antara faktor analisis dan path analysis menjadi satu metodologi statistik yang komprehensif.

Model Persamaan Struktural atau Structural Equation Modeling (SEM) adalah sekumpulan metode-metode statistika yang memungkinkan pengujian suatu rangkaian hubungan yang relatif kompleks secara simultan.

#### **2.2.4.2 Model Atau Elemen Dalam SEM**

##### **a. Model Struktural (*Structural Model*)**

Model struktural merupakan model yang menggambarkan hubungan antar variabel laten.

##### **b. Model pengukuran (*Measurement Model*)**

Setiap variabel laten mempunyai beberapa ukuran atau variabel teramati atau indikator. Variabel laten dihubungkan dengan variabel teramati melalui model pengukuran yang dibentuk analisis faktor. Setiap variabel laten dimodelkan sebagai faktor yang mendasari variabel terkait.

#### **2.2.4.3 Proses Pemodelan SEM**

Pemodelan SEM yang lengkap pada dasarnya terdiri dari Measurement Model dan Structural Model. Measurement Model ditujukan untuk mengkonfirmasi dimensi-dimensi

yang dikembangkan pada sebuah faktor, sedangkan Structural Model mengenai stuktur hubungan yang membentuk atau menjelaskan kausalitas antar faktor.

Hair dan Anderson menyatakan, tujuh tahapan dalam SEM yaitu: (1) Pengembangan Model Berbasis Teori, (2) Pengembangan Diagram Jalur (Menunjukkan Hubungan Kausalitas), (3) Konversi Diagram Jalur (Serangkaian Persamaan Struktural & Spesifikasi Model), (4) Pemilihan Matriks Input & Teknik Estimasi Atas Model Yang Dibangun, (5) Pemeriksaan Masalah Identifikasi, (6) Evaluasi Model, (7) Interpretasi & Modifikasi Model [18].

#### **2.2.4.4 SEM Berbasis Component atau variance-GSCA**

Menurut peneliti Wold pada tahun 1985, GSCA merupakan metode analisis yang powerfull oleh karena tidak didasarkan banyak asumsi [19]. Data tidak harus berdistribusi normal multivariate (Indikator dengan skala kategori, ordinal, interval sampai ratio dapat digunakan pada model yang sama), sample pun tidak harus besar. Berikut Tabel 2.1 merupakan ringkasan perbandingan antara SEM berbasis variance-GSCA dengan sem berbasis covariance [20].

Pada awalnya component based SEM dilakukan dengan menggunakan PLS, namun PLS mempunyai beberapa kelemahan. Salah satu kelemahannya yaitu, PLS tidak memberikan mekanisme untuk menilai overall goodness-fit dari model. Tidak adanya ukuran goodness-fit model menjadi sulit menentukan seberapa baik model fit (cocok) dengan datanya dan sulit untuk membandingkan dengan model alternatif [20].

Untuk mengatasi kelemahan PLS tersebut, maka dikembangkanlah metode Generalized Structured Component Analysis (GSCA). GSCA merupakan bagian dari component based SEM yang menawarkan criteria global least square optimization, yang secara konsisten meminimumkan untuk mendapatkan estimasi parameter model [20].

**Tabel 2.1 Perbandingan antara SEM berbasis variance-GSCA dengan sem berbasis covariance**

<b>Kriteria</b>	<b>CB-SEM</b>	<b>GSCA</b>
Tujuan	Untuk menguji teori atau mengkonfirmasi teori (Orientasi Parameter)	Untuk mengembangkan atau membangun teori (orientasi prediksi)
Pendekatan	Berdasarkan covariance	Berdasarkan variance
Spesifikasi Model Pengukuran	Menyaratkan adanya error terms dan indicator hanya berbentuk reflective.	Indikator dapat berbentuk reflective dan formative
Model Structural	Model dapat berbentuk recursive dan non-recursive dengan tingkat kompleksitas kecil sampai menengah	Model dengan kompleksitas Besar dengan banyak konstruk dan banyak indicator
Karakteristik Data dan Alogarithm	Mensyaratkan jumlah sample yang besar dan asumsi multivariate normality terpenuhi (parametik)	Jumlah Sample dapat kecil dan tidak mensyaratkan asumsi multivariate normality (non-parametik)
Evaluasi Model	Mensyaratkan terpenuhinya criteria goodness of fit sebelum estimasi parameter	Mensyaratkan terpenuhinya criteria goodness of fit untuk evaluasi model
Pengujian Signifikansi	Model dapat diuji dan difalsifikasi	Tidak dapat diuji dan difalsifikasi
Software Error	Sering bermasalah dengan inadmissible dan faktor indeterminacy	Sering bermasalah dengan inadmissible dan faktor indeterminacy
Implikasi	Optimal untuk ketepatan parameter	Optimal untuk ketepatan prediksi

GSCA merupakan bagian dari component based SEM yang menawarkan criteria global least square optimization, yang secara konsisten meminimumkan untuk mendapatkan estimasi parameter model [20]. Selain itu, GSCA juga dilengkapi dengan ukuran model fit secara keseluruhan.

Keunggulan lain yang dapat dari menggunakan metode GSCA adalah sebagai berikut:

1. Tidak memerlukan asumsi distribusi tertentu, misalnya distribusi normalitas.
2. Bisa menggunakan indikator formatif maupun reflektif secara bersamaan.
3. Dapat menghindari adanya faktor determinasi.

Dalam penelitian ini, perangkat lunak yang akan digunakan untuk melakukan analisis adalah aplikasi GeSCA. Aplikasi ini merupakan aplikasi berbasis website yang diciptakan oleh Prof. Heunsun Hwang dari McGill University. Aplikasi ini GeSCA ini dapat diakses secara online melalui alamat [www.sem-gesca.org](http://www.sem-gesca.org). Kemudahan yang didapat dari menggunakan GeSCA antara lain adalah minimal sampel yang digunakan hanya 30 sampel, dan semua perhitungan yang diperlukan hanya membutuhkan satu kali *running*. Dalam sekali *running* itu juga ditampilkan AVE dan Alpha.

#### **2.2.4.5 Generalized Structured Component Analysis**

GSCA menawarkan criteria global least square optimization, yang secara konsisten meminimumkan untuk mendapatkan estimasi parameter model. GSCA juga dilengkapi dengan ukuran model fit secara keseluruhan. Dalam SEM berbasis variance ini nantinya akan dijumpai 2 model indikator yaitu refleksif dan formatif.

Untuk model indikator refleksif, indikator-indikator pada suatu konstruk (variable laten) dipengaruhi oleh konsep yang sama. Perubahan dalam suatu item atau indikator akan berakibat pada perubahan indikator lainnya dengan arah yang sama. Sedangkan dalam model formatif, indikator dipandang sebagai variable yang mempengaruhi variabel laten. Sebagai misal Cohen et al., menggunakan variabel laten status sosial ekonomi (SSE) dengan indikator-indikator antara lain pendidikan, prestige, pekerjaan dan pendapatan [21]. Dalam hal ini indikator pendidikan, prestise pekerjaan dan

pendapatan mempengaruhi variabel laten status sosial ekonomi. Dimana jika salah satu indikator meningkat maka indikator yang lain tidak harus ikut meningkat pula. Kenaikan pada satu indikator pendapatan akan meningkatkan variabel laten status sosial ekonomi (SSE).

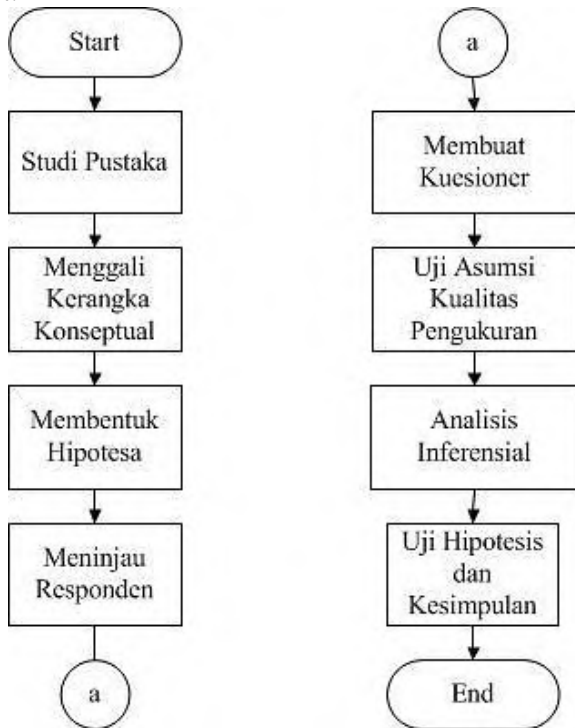
GSCA dapat dipandang sebagai component based SEM dimana variabel laten didefinisikan sebagai komponen tertimbang dari variable observed. GSCA meliputi juga model pengukuran (measurement model) yang menggambarkan hubungan antara indikator dan konstruk. Serta model structural yang menghubungkan antara konstruk. GSCA memberikan ukuran overall model fit yang disebut FIT. Nilai FIT berkisar dari 0 sampai 1.

Semakin besar nilai FIT, maka semakin besar variance dari data yang dapat dijelaskan oleh model. Namun demikian nilai FIT dipengaruhi oleh kompleksitas model sehingga dikembangkan Adjusted FIT (AFIT) yang telah memasukkan kompleksitas model. Derajat-bebas (degree of freedom) untuk null model ( $W=0$  dan  $A=0$ ) dan yang merupakan derajat bebas model yang diuji, dan parameter bebas. GSCA juga memberikan 2 tambahan ukuran model fit: yang pertama unweighted least square GFI dan yang kedua SMSR (standardize root mean square residual). Nilai GFI mendekati 1 dan SRMR mendekati 0 merupakan indikasi good fit [22].



### BAB III METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan metode yang digunakan dalam pengerjaan tugas akhir. Metode ini digunakan agar pengerjaan tugas akhir dapat dilakukan dengan sistematis. Tahapan pengerjaan tugas akhir dapat dilihat pada gambar 3.1 berikut.



**Gambar 3.1 Metodologi Pengerjaan Tugas Akhir**

Melalui Gambar 3.1, dapat diketahui langkah-langkah pengerjaan tugas akhir yang akan dilakukan yaitu sebagai berikut:

### **3.1 Studi Pustaka**

Studi literatur perlu dilakukan sebagai dasar dalam melakukan tugas akhir. Studi literatur dilakukan dengan mencari sumber-sumber pendukung. Sumber-sumber yang dimaksud dapat berupa jurnal tugas akhir, buku, e-book yang membahas tentang pembuatan kuesioner yang disesuaikan antara responden dengan informasi yang akan dicari nantinya. Kuesioner dan model konseptual merupakan adopsi dari paper. Referensi lain yang terkait dengan pengerjaan tugas akhir ini adalah teori-teori dalam analisa statistika dan SEM yang digunakan untuk mengolah data yang telah terkumpul dari hasil kuesioner.

### **3.2 Penggalan Kerangka Konseptual**

Kerangka berfikir atau juga disebut kerangka konseptual merupakan model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi sebagai masalah yang penting [23]. Suatu kerangka berfikir akan memberikan penjelasan sementara terhadap gejala yang menjadi masalah atau objek penelitian. Tahapan ini bertujuan untuk memberikan penjelasan sementara terhadap gejala yang menjadi masalah/objek penelitian.

### **3.3 Pembentukan Hipotesa**

Setelah mendapatkan kerangka konseptual, maka dilakukan pembentukan hipotesis untuk membentuk dugaan sementara pada kerangka konseptual. Dugaan sementara ini berasal dari hasil analisis nilai Effort Rate (ER) pada penelitian yang lalu, yang akan dibuktikan pada langkah-langkah setelah pembentukan hipotesis [9].

### **3.4 Peninjauan Responden**

Data yang akan dianalisis dalam penelitian ini diambil secara langsung (primer) dengan mengajukan pertanyaan

melalui kuisioner. Responden yang digali informasinya adalah perwakilan dari proyek pengembangan perangkat lunak dan disebarkan secara merata berdasarkan urutan nilai *Effort Rate* atau *Productivity Factor*.

### **3.5 Pembuatan Kuesioner**

Pada tahapan ini, peneliti merancang kuesioner sesuai dengan kebutuhan dan tujuan yang diinginkan. Beberapa hal yang nantinya harus diperhatikan saat merancang kuesioner adalah menetapkan informasi yang ingin diketahui, menentukan jenis kuesioner dan metode administrasinya, menentukan isi dari masing-masing pertanyaan, menentukan banyak respon atas setiap pertanyaan, menentukan kata-kata yang digunakan untuk setiap pertanyaan, menentukan urutan pertanyaan, menentukan karakteristik fisik kuesioner serta melakukan uji awal atas kuesioner.

### **3.6 Uji Validitas dan Uji Reliabilitas**

Proses pada tahapan ini adalah melakukan uji validitas terhadap data dari kuesioner yang telah didapatkan. Jika data tersebut belum valid, langkah yang dilakukan adalah menghapus beberapa pertanyaan yang belum valid. Setelah hasil data yang didapatkan adalah valid, maka dapat dilakukan uji selanjutnya.

Berikutnya adalah uji reliabilitas, yaitu untuk memastikan apakah kuesioner penelitian yang akan dipergunakan untuk mengumpulkan data reliabel atau tidak. Kuesioner dikatakan reliabel jika kuesioner tersebut dilakukan pengukuran berulang, akan mendapatkan hasil yang sama.

### **3.7 Analisis Inferensial**

Analisis Inferensial dilakukan dengan menggunakan pendekatan berbasis komponen, yakni dengan alat bantu *Generalized Structured Component Analysis* (GSCA). Analisis ini dimulai dari melakukan perancangan model

struktural, melakukan perancangan model pengukuran, mengembangkan diagram jalur. Setelah output dari GSCA keluar, dilakukan estimasi terhadap *weight and loading estimate*, *path coefficient estimate*, lalu *means weight*, *loading* dan *path coefficient*.

### **3.8 Uji Hipotesis**

Setelah semua proses dilakukan, pada tahapan ini didapatkan hasil. Hasil bisa berupa kesesuaian dengan hipotesis di awal ataupun adanya hasil di luar hipotesis tersebut. Dari hasil uji ini, peneliti akan mendapatkan jawaban atas rumusan masalah.

### **3.9 Kesimpulan**

Tahapan terakhir yang dilakukan adalah membuat pembuatan laporan dan simpulan dari hasil penelitian yang telah dilakukan. Simpulan ini menjawab rumusan masalah yang telah ditentukan di awal.

Dari simpulan tersebut, selanjutnya diajukan saran baik yang ditujukan bagi pengembang perangkat lunak maupun untuk penelitian selanjutnya. Pembuatan laporan di sini ditujukan agar seluruh langkah-langkah yang telah dilakukan didokumentasikan dengan lengkap sehingga dapat memberikan informasi yang berguna bagi yang membacanya.

## **BAB IV PERANCANGAN**

Pada bab ini akan dijelaskan mengenai bagaimana penelitian akan dilakukan, pemilihan obyek penelitian, dan strategi pelaksanaan, dari konsep hingga kuesioner terbentuk

### **4.1 Penggalian Kerangka Konseptual**

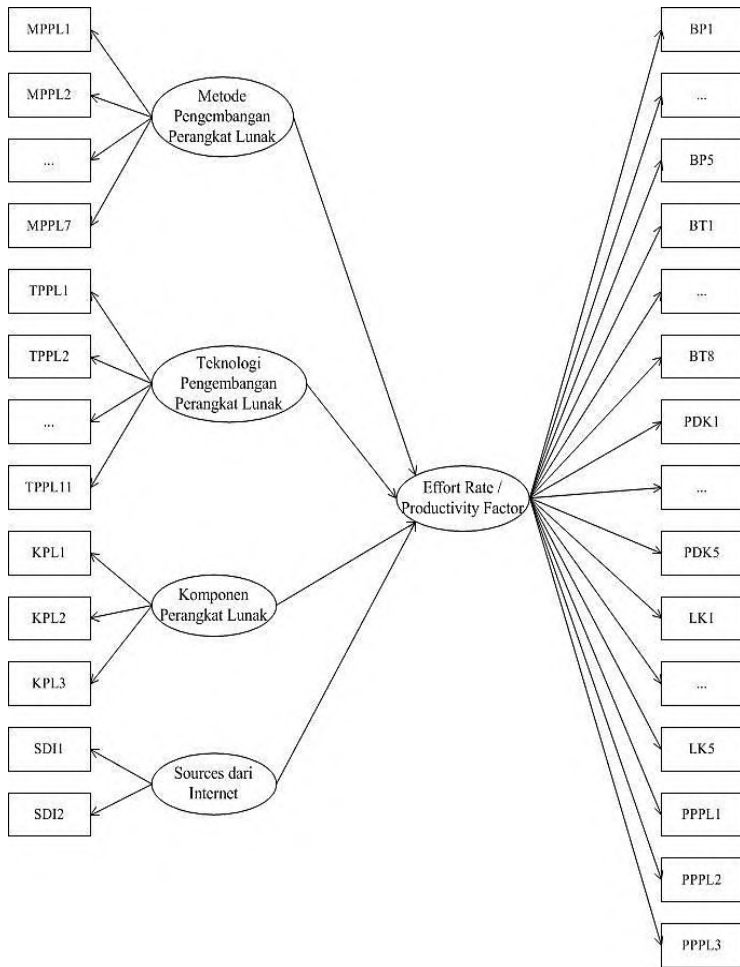
Kerangka berfikir atau juga disebut kerangka konseptual merupakan model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi sebagai masalah yang penting [23]. Suatu kerangka berfikir akan memberikan penjelasan sementara terhadap gejala yang menjadi masalah atau objek penelitian. Tahapan ini bertujuan untuk memberikan penjelasan sementara terhadap gejala yang menjadi masalah/objek penelitian.

Output dari tahapan ini adalah terbentuknya sebuah kerangka konseptual yang dibutuhkan dalam penelitian ini. Berdasarkan teori dasar dan penelitian terdahulu mengenai nilai Effort Rate (ER) [9], maka pola hubungan masing – masing faktor dapat digambarkan pada Gambar 4.1.

Konseptual model pada gambar 4.1 disusun dari hasil analisis nilai Effort Rate (ER) pada penelitian terdahulu dengan menggunakan metode UCP [9]. Masing – masing faktor dugaan di atas memiliki penjelasan sebagai berikut:

#### **1. Metode Pengembangan Perangkat Lunak**

Metode pengembangan perangkat lunak atau *software engineering* memberikan teknik-teknik bagaimana mengembangkan perangkat lunak. Terdiri dari serangkaian tugas (*task*) seperti perencanaan dan estimasi proyek.



**Gambar 4.1 Konseptual Model**

Ada banyak macam model dalam metode pengembangan perangkat lunak, yaitu *waterfall*, *prototyping*, *spiral* dan lain-lain.

## 2. Teknologi Pengembangan Perangkat Lunak

Semakin berkembangnya teknologi informasi maka semakin berkembang pula tool untuk membantu dalam membangun perangkat lunak. Jika dahulu programmer mengetik baris-baris program dalam console atau terminal, kini banyak tersedia *software tool* pembuatan program sesuai bahasa pemrograman yang dibutuhkan. Selain itu, kini software developer dapat mengetik baris program, mendesain tampilan (interface), mengkompilasinya (compile), dan kemudian membuat file executable (.exe) serta mengkoneksikannya dengan database dalam satu *development tool software*.

### 3. Komponen Perangkat Lunak

Saat ini para *software developer* dalam membuat atau mengembangkan perangkat lunak dimudahkan juga dengan menggunakan komponen-komponen yang telah ada, seperti kumpulan *library* pada *framework*. Komponen tersusun atas kelas-kelas (*class*) yang dapat digunakan kembali (*re-usable*) dan bersifat iteratif yang disusun atau diorganisasikan sesuai kebutuhan pengembangan.

### 4. Sources Dari Internet

Berbagai literatur bisa didapatkan dari internet. Selain itu *source code* juga dapat dicari di internet, hal tersebut dapat memudahkan *software developer* dalam hal melakukan tiap kegiatan pengembangan perangkat lunak. Juga terdapat referensi dan contoh proyek perangkat lunak yang serupa atau memiliki jenis yang sama dengan perangkat lunak yang sedang dikembangkan, sehingga dapat dijadikan acuan dalam pengembangan perangkat lunak dan dapat menghemat waktu pengerjaan.

### 5. *Effort Rate* atau *Productivity Factor*

Menerjemahkan *Use Case Point* menjadi *man-hours* atau *person-hours* per UCP adalah soal menghitung standar

penggunaan *Effort Rate* (ER) dan mengalikan nilai ER tersebut dengan jumlah UCPs. data historis milik Tim studi kasus menetapkan *Effort Rate* proyek di 28 *person-hours* per UCP [24].

Selain itu, pengalaman mereka sependapat dengan Schneider dan Winters [6], bahwa tidak semua proyek sama, sehingga mereka membedakan proyek yang sederhana dari proyek yang kompleks dengan menetapkan 20 *person-hours* per UCP pada proyek yang sederhana.

Ketika produktivitas dimasukkan sebagai koefisien yang mengungkapkan waktu, persamaannya dapat digunakan untuk memperkirakan jumlah *person-hours* yang dibutuhkan untuk menyelesaikan proyek. Berikut ini adalah persamaan lengkapnya dengan memasukkan *Productivity Factor* (PF) [7]:

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{ECF} * \text{PF} \dots\dots\dots(1)$$

PF mendefinisikan *Productivity Factor* di kisaran 20 sampai 40 *person-hours* per UCP dan harus dikalibrasikan dengan produktivitas organisasi yang sesuai. *The estimated project effort* menurut Karner didefinisikan sebagai [25],

$$\text{PE}_{\text{Karner}} := \text{UCP} * \text{PF} \dots\dots\dots(2)$$

Keterangan:

PE = *The Estimated Project Effort*

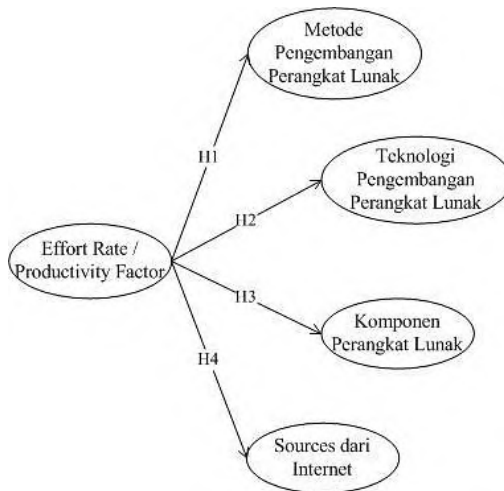
UCP = *Use Case Point*

PF = *Productivity Factor*

## 4.2 Pembentukan Hipotesa

Output dari pembentukan hipotesis adalah memperoleh beberapa hipotesis-hipotesis yang dijadikan tujuan dalam melakukan penelitian ini. Hipotesis dibentuk dari dugaan sementara pada model konseptual di bawah ini:





**Gambar 4.2 Acuan Hipotesis**

Dengan mengacu pada model konseptual di atas, maka hipotesis yang akan digunakan dalam penelitian adalah seperti dibawah ini:

**H1: Metode Pengembangan Perangkat Lunak berpengaruh pada *Productivity Factor*, sehingga Nilai ER / PF mengalami penurunan.**

*Software developer* membutuhkan metode pengembangan PL yang akan memudahkan pengembang proyek perangkat lunak dalam mengakomodasi beberapa kebutuhan. Dengan metode pengembangan PL, pekerjaan dapat dilakukan dalam pola yang tepat, produk dicapai dalam waktu yang telah ditentukan dan biaya menjadi efektif [9].

Kesesuaian metode ini akan berpengaruh terhadap perubahan nilai *productivity factor*. Semakin sesuai metode pengembangan, maka produktivitas tim *developer* akan semakin tinggi, dan akan menurunkan nilai *effort rate* atau *productivity factor*.

**H2: Teknologi Pengembangan Perangkat Lunak berpengaruh pada *Productivity Factor*, sehingga Nilai ER / PF mengalami penurunan.**

Telah banyak *software tool* yang dapat membantu dalam pengembangan perangkat lunak. Kegiatan-kegiatan dalam pengembangan perangkat lunak seperti, membuat rencana proyek, analisa kebutuhan, desain tampilan dan basis data, coding, dan membuat database dapat dimudahkan karena kemajuan teknologi *software engineering*.

Kecanggihan *software tool* ini akan berpengaruh terhadap perubahan nilai *productivity factor*. Semakin canggih teknologi pengembangan, maka pekerjaan tim *developer* akan semakin dipermudah, dan akan menurunkan nilai *effort rate* atau *productivity factor*.

**H3: Komponen Perangkat Lunak berpengaruh pada *Productivity Factor*, sehingga Nilai ER / PF mengalami penurunan.**

Penggunaan komponen *software* yang telah ada dapat menguntungkan dari segi siklus waktu pengembang perangkat lunak, karena mampu mengurangi waktu sebesar 70% serta biaya produksi berkurang sampai 84% arena pembangunan komponen berkurang [9].

Sebagai contoh, Microsoft Visual Basic versi 5.0 telah menyediakan teknologi komponen Microsoft ActiveX yang bertujuan untuk digunakan di internet, intranet dan lingkungan tradisional client.

Penggunaan komponen *software*, seperti *class* atau *library* ini akan berpengaruh terhadap perubahan nilai *productivity factor*. Semakin banyak komponen yang terlibat, maka akan menurunkan nilai *effort rate* atau *productivity factor*.

**H4: Sumber dari internet berpengaruh pada *Productivity Factor*, sehingga Nilai ER / PF mengalami penurunan.**

Kesesuaian *source code* yang didapatkan dari internet, atau contoh proyek perangkat lunak yang serupa akan berpengaruh terhadap perubahan nilai *productivity factor*. Semakin banyak referensi yang dijadikan rujukan, maka produktivitas tim *developer* akan semakin tinggi, dan akan menurunkan nilai *effort rate* atau *productivity factor*.

### 4.3 Pembuatan Kuesioner

#### 4.3.1 Prosedur Analisa

Prosedur analisa yang digunakan pada penelitian ini adalah berdasarkan metode *Use Case Point*. Dari metode Use Case Point terdapat rumus yang diturunkan menjadi indikator beberapa area penilaian yaitu, area metode pengembangan perangkat lunak, area teknologi pengembangan perangkat lunak, area komponen perangkat lunak, dan area *sources* dari internet.

Pada metode Use Case Point terdapat faktor teknikal yang mengacu pada faktor-faktor *Technical Complexity Adjustment* yang terdapat pada metode *Function Point Analysis*, dan faktor-faktor lingkungan yang digunakan untuk menghitung fungsi-fungsi yang tidak fungsional yang biasa digunakan untuk mempermudah pekerjaan seorang programmer [26].

Faktor-faktor tersebut memiliki bobot dan nilai atau skor. Nilai atau skor yang diberikan pada setiap faktor, tergantung dari seberapa besar pengaruh dari faktor tersebut.

1. 0 = Menandakan tingkat paling rendah atau tidak mempengaruhi,
2. 3 = Berarti rata-rata, hingga
3. 5 = Menandakan tingkat paling maksimal atau memberikan pengaruh yang besar.

Sedangkan bobot untuk masing-masing faktor tersebut telah didefinisikan oleh Gustav Karner.

Dari *Technical Complexity Factor* dan *Technical Complexity Factor* kemudian diturunkan menjadi indikator untuk area metode pengembangan perangkat lunak, area teknologi pengembangan perangkat lunak, area komponen perangkat lunak, dan area *sources* dari internet.

#### 4.3.1.1 Technical Complexity Factor (TCF)

*Technical Complexity Factor* adalah salah satu faktor yang diterapkan untuk memperkirakan ukuran software dengan menghitung Pertimbangan Teknis dari sebuah sistem. Nilai yang diberikan pada setiap faktor ditentukan dengan angka antara 0 (faktor tidak relevan) dan 5 (faktor relevan). Berikut adalah 13 *technical factors* / Faktor Teknikal yang ditunjukkan pada Tabel 4.1,

**Tabel 4.1 Technical Factor dan bobotnya**

<b>Ti</b>	<b>Technical Factor</b>	<b>Weight</b>
T1	Required Distributed Systems	2
T2	Response Time Is Important	1
T3	End User Efficiency	1
T4	Required Complex Internal Processing	1
T5	Reusable Code to Focus	1
T6	Installation Easy	0,5
T7	Usability	0,5
T8	Cross-Platform Support	2
T9	Easy To Change	1
T10	Highly Concurrent	1
T11	Custom Security	1
T12	Dependence On Third-Part Code	1
T13	User Training	1

#### 4.3.1.2 Environmental Complexity Factor (ECF)

*Environmental Complexity Factor* adalah salah satu faktor yang diterapkan untuk memperkirakan ukuran software dengan menghitung Pertimbangan Lingkungan Sekitar dari sebuah sistem. Nilai yang diberikan pada setiap faktor ditentukan dengan angka antara 0 (faktor tidak relevan) dan 5 (faktor relevan). Berikut adalah 8 *environmental factors* / Faktor Lingkungan Sekitar yang ditunjukkan pada tabel di bawah,

**Tabel 4.2 Environmental Factor dan bobotnya**

<b>Ei</b>	<b>Environmental Factor</b>	<b>Weight</b>
E1	Familiariaty with the Project	1,5
E2	Application Experience	0,5
E3	OO Programming Experience	1
E4	Lead Analyst Capability	0,5
E5	Motivation	1
E6	Stable Requirements	2
E7	Part Time Staff	-1
E8	Difficulty Programming Language	-1

#### 4.3.1.3 Metode Pengembangan Perangkat Lunak

Pada area ini penilaian difokuskan pada hal-hal yang berhubungan dengan metode pengembangan, pengalaman pengembang, dan motivasi di dalam tim. Indikator penilaian pada metode pengembangan perangkat lunak adalah sebagai berikut:

**Tabel 4.3 Indikator Metode Pengembangan Perangkat Lunak**

<b>Indikator</b>	<b>Deskripsi</b>
<b><i>Required Distributed Systems</i></b>	Tingkat kompleksitas rancangan pada metode pengembangan perangkat lunak

<b>Indikator</b>	<b>Deskripsi</b>
<b><i>Required Complex Internal Processing</i></b>	Tingkat kompleksitas algoritma pada metode pengembangan perangkat lunak
<b><i>Familiarity with the Method</i></b>	Tingkat penguasaan metode pengembangan perangkat lunak pada pengembang
<b><i>OO Programming Experience</i></b>	Tingkat pengalaman pengembang pada Object Oriented (OO) programming
<b><i>Motivation</i></b>	Tingkat motivasi pengembang dalam menggunakan berbagai metode pengembangan perangkat lunak
<b><i>Highly Concurrent</i></b>	Tingkat perselisihan dalam memilih metode pengembangan perangkat lunak
<b><i>Analyst Capability</i></b>	Tingkat kemampuan analisa dalam tim untuk memilih metode pengembangan perangkat lunak
<b><i>Part Time Staff</i></b>	Tingkat pemanfaatan anggota tim yang bekerja paruh waktu

#### **4.3.1.4 Teknologi Pengembangan Perangkat Lunak**

Pada area ini penilaian difokuskan pada hal-hal yang berhubungan dengan teknologi aplikasi pengembangan, pengalaman pengembang, dan motivasi di dalam tim. Indikator penilaian pada teknologi pengembangan perangkat lunak adalah sebagai berikut:

**Tabel 4.4 Indikator Teknologi Pengembangan Perangkat Lunak**

<b>Indikator</b>	<b>Deskripsi</b>
<b><i>Response Time Is Important</i></b>	Pentingnya kecepatan respon (server) aplikasi pengembang perangkat lunak
<b><i>End User Efficiency</i></b>	Tingkat optimalisasi dalam menggunakan aplikasi pengembang perangkat lunak
<b><i>Installation Easy</i></b>	Tingkat kemudahan instalasi aplikasi pengembang perangkat lunak

<b>Indikator</b>	<b>Deskripsi</b>
<i>Usability</i>	Tingkat kemudahan penggunaan aplikasi pengembang perangkat lunak
<i>Cross-Platform Support</i>	Aplikasi pengembang perangkat lunak mendukung multi-platform
<i>Easy To Change</i>	Tingkat kemudahan kustomisasi aplikasi pengembang perangkat lunak
<i>Custom Security</i>	Tingkat keamanan pada aplikasi pengembangan perangkat lunak
<i>User Training</i>	Lama waktu pelatihan pengembang untuk menggunakan aplikasi pengembang perangkat lunak
<i>Motivation</i>	Tingkat motivasi pengembang untuk menggunakan teknologi yang dimiliki aplikasi pengembang perangkat lunak
<i>Difficulty Programming Language</i>	Tingkat kesulitan bahasa pemrograman yang digunakan pada aplikasi pengembangan perangkat lunak
<i>Application Experience</i>	Tingkat pengalaman pengembang dalam memakai aplikasi pengembangan perangkat lunak

#### 4.3.1.5 Komponen Perangkat Lunak

Pada area ini penilaian difokuskan pada hal-hal yang berhubungan dengan pemanfaatan komponen perangkat lunak. Indikator penilaian pada komponen perangkat lunak adalah sebagai berikut:

**Tabel 4.5 Indikator Komponen Perangkat Lunak**

<b>Indikator</b>	<b>Deskripsi</b>
<i>Reusable Code Must Be A Focus</i>	Tingkat penggunaan ulang kode yang sudah ada dalam mengembangkan perangkat lunak
<i>Dependence On Third-Party Code</i>	Tingkat penggunaan Third-Party code dalam mengembangkan perangkat lunak
<i>Stable Requirements</i>	Tingkat perubahan kebutuhan dalam mengembangkan perangkat lunak

#### 4.3.1.6 Sources dari Internet.

Pada area ini penilaian difokuskan pada hal-hal yang berhubungan dengan pemanfaatan referensi yang berasal dari internet. Indikator penilaian pada *sources dari internet* adalah sebagai berikut:

**Tabel 4.6 Sources dari Internet**

<b>Indikator</b>	<b>Deskripsi</b>
<b><i>Reusable Code Must Be A Focus</i></b>	Tingkat penggunaan ulang kode yang berasal dari internet dalam mengembangkan perangkat lunak
<b><i>Dependence On Third-Party Code</i></b>	Tingkat penggunaan Third-Party code yang berasal dari internet dalam mengembangkan perangkat lunak

#### 4.3.1.7 Productivity Factor

Penilaian pada area ini berdasarkan jurnal “A Systematic Review of Productivity Factors in Software Development” oleh “Stefan Wagner dan Melanie Ruhe” [27]. Dari jurnal tersebut, terdapat tabel yang berhubungan dengan *soft factors* yang diturunkan menjadi indikator area penilaian yaitu, area *productivity factor* [27].

*Productivity factor* terbagi menjadi *technical factors* dan *soft factors*. *soft factors* adalah semua faktor non teknis yang mempengaruhi produktivitas. Faktor – faktor ini berasal dari tim pengembang dan lingkungan kerja tim [27].

*The soft factors* diringkas ke dalam tabel 4.7. Faktor - faktor yang tumpang tindih dikombinasikan sejauh mungkin. Dengan menggunakan kategorisasi yang non-unik dan sederhana untuk membantu pemahaman yang cepat. Kategori *corporate cultures* (budaya perusahaan) mengandung faktor – faktor yang berada pada tingkat perusahaan, sedangkan kategori *team cultures* (budaya tim) menunjukkan faktor yang



serupa yang berada di tingkat regu. Pada kategori *capabilities and experiences* (pengalaman dan kemampuan) adalah faktor – faktor yang berhubungan lebih kepada individu. Kategori *environment* (lingkungan kerja) adalah faktor dari properti dan keadaan lingkungan kerja. Terakhir, faktor yang lebih spesifik ke arah proyek ada di dalam kategori *project* (proyek pengembangan perangkat lunak) [27].

**Tabel 4.7 The Soft Factors**

<b>Factor</b>	<b>Description</b>
<b>Corporate Culture</b>	
Credibility	Open communication and competent organisation
Respect	Opportunities and responsibilities
Fairness	Fairness in compensation and diversity
<b>Team Culture</b>	
Camaraderie	Social and friendly atmosphere in the team
Team Identity	The common identity of the team members
Sense of Eliteness	The feeling in the team that they are "superior"
Clear Goals	How clearly defined are the group goals?
Turnover	The amount of change in the personnel
Team Cohesion	The cooperativeness of the stakeholders
Communication	The degree and efficiency of which information flows in the team
Support for Innovation	To what degree assistance for new ideas is available
<b>Capabilities and Experiences</b>	
Developer Temperaments	The mix of different temperaments on the team
Programmer Capability	The skills of the programmer

<b>Factor</b>	<b>Description</b>
Platform Experience	The familiarity with the hard and software platform
Manager Capability	The control of the manager over the project
Manager Application Experience	The familiarity of the manager with the application
<b>Environment</b>	
Proper Workplace	The suitability of the workplace to do creative work
E-Factor	This environmental factor describes the ratio of uninterrupted hours and body present hours
Time Fragmention	The amount of necessary "context switches" of an employee
Physical Separation	The team members are distributed over the building or multiple sites
Telecommunication Facilities	Support for work at home, virtual teams, video conferencing with clients
<b>Project</b>	
Schedule	The appropriateness of the schedule for the development task
Requirements Stability	The number of requirements changes
Average Team Size	Number of people in the team

Indikator penilaian pada *productivity factor* adalah sebagai berikut:

**Tabel 4.8 Productivity Factor**

<b>Indikator</b>	<b>Deskripsi</b>
<b>Budaya Perusahaan</b>	
Kredibilitas	Kinerja dari sebuah organisasi dan keterbukaan dalam berkomunikasi
Kehormatan	Tingkat peluang dan tanggung jawab yang diberikan organisasi

<b>Indikator</b>	<b>Deskripsi</b>
Keadilan	Tingkat keadilan sebuah organisasi dalam memberikan imbalan
<b>Budaya Tim</b>	
Persahabatan	Tingkat keramahan tim dalam bersosialisasi
Jati Diri Tim	Tingkat kejelasan identitas anggota tim
Perasaan Elite	Tingkat perasaan unggul di dalam tim
Sasaran yang Jelas	Tingkat kejelasan sasaran di dalam tim
Perputaran	Tingkat pergantian anggota tim
Hubungan Erat	Tingkat kooperatif dari para pemangku kepentingan
Komunikasi	Tingkat efisiensi informasi yang mengalir di dalam tim
Dukungan Inovasi	Tingkat bantuan untuk ide-ide baru
<b>Pengalaman Dan Kemampuan</b>	
Gaya Perilaku Pengembang	Tingkat perbedaan perilaku anggota di dalam tim
Kemampuan Programmer	Tingkat kemampuan programmer di dalam tim
Pengalaman Platform	Tingkat kemahiran dengan platform perangkat keras maupun lunak
Kemampuan Manajer	Tingkat kendali manajer terhadap proyek
Pengalaman Manajer terhadap Aplikasi	Tingkat kemahiran manajer dengan aplikasi
<b>Lingkungan Kerja</b>	
Kelayakan Workplace	Keserasian workplace untuk mengerjakan karya kreatif

<b>Indikator</b>	<b>Deskripsi</b>
E-Factor	Tingkat kebisingan dalam lingkungan pengembangan perangkat lunak
Fragmentasi Waktu	Jumlah "context switches" yang diperlukan seorang anggota tim
Pemisahan secara Fisik	Pendistribusian anggota tim pada satu workplace atau beberapa workplaces
Fasilitas Telekomunikasi	Dukungan untuk bekerja di rumah, membentuk tim virtual, atau konferensi video dengan klien
<b>Proyek Pengembangan Perangkat Lunak</b>	
Penjadwalan	Kelayakan jadwal untuk pekerjaan pengembangan perangkat lunak
Stabilitas kebutuhan	Jumlah dari perubahan kebutuhan
Rata-rata besar tim	Jumlah orang di dalam tim

#### 4.3.2 Instrumen Penelitian

Pada penelitian ini digunakan instrumen penelitian yang berupa kuesioner dan mengacu pada faktor dan indikator yang telah diturunkan dari metode Use Case Point dan jurnal “*A Systematic Review of Productivity Factors in Software Development*”. Pilihan jawaban kuesioner menggunakan skala likert dengan nilai pengukuran sebagai berikut:

- Nilai 1 = Sangat Tidak Setuju
- Nilai 2 = Tidak Setuju
- Nilai 3 = Ragu-ragu
- Nilai 4 = Setuju
- Nilai 5 = Sangat Setuju

Kuesioner lebih lengkapnya dapat dilihat pada Lampiran A.

#### 4.3.2.1 Daftar Pertanyaan Kuesioner

Penyusunan pertanyaan kuesioner harus berdasarkan 5 area penelitian dan indikator – indikator yang telah diturunkan dari metode Use Case Point dan jurnal “*A Systematic Review of Productivity Factors in Software Development*”. Bentuk dari pertanyaan kuesioner yang dibuat adalah sebagai berikut:

**Tabel 4.9 Pertanyaan Tentang Metode Pengembangan Perangkat Lunak**

Kode	No	Pernyataan
MPPL	1	Kebutuhan membuat rancangan dalam metode pengembangan perangkat lunak sangat kompleks. (Semakin kompleks metode pengembangan, maka semakin sangat setuju).
	2	Banyak algoritma pengolahan data yang efektif pada metode pengembangan perangkat lunak yang digunakan. (Semakin efektif metode pengembangan, maka semakin sangat setuju).
	3	Tim anda merasa familiar (menguasai) dengan metode pengembangan perangkat lunak yang sedang digunakan. (Semakin familiar, maka semakin sangat setuju).
	4	Tim anda paham dengan Pemrograman berorientasi objek atau <i>Object Oriented (OO) programming</i> . (Semakin paham dengan Pemrograman berorientasi objek, maka semakin sangat setuju).

Kode	No	Pernyataan
	5	Tim anda termotivasi dengan metode pengembangan perangkat lunak yang sedang digunakan. (Semakin besar motivasi dalam tim, maka semakin sangat setuju).
	6	Tim anda sering berselisih pada saat memilih metode pengembangan perangkat lunak. (Semakin tim sering berselisih, maka semakin sangat setuju).
	7	Tim anda mampu menganalisa metode pengembangan perangkat lunak mana yang cocok untuk digunakan. (Semakin besar kapabilitas dan pengetahuan tim, maka semakin sangat setuju).
	8	Di dalam tim anda terdapat anggota tim yang bekerja paruh waktu. (Semakin banyak anggota tim untuk bekerja paruh waktu, maka semakin sangat setuju).

**Tabel 4.10 Pertanyaan Tentang Teknologi Pengembangan Perangkat Lunak**

Kode	No	Pernyataan
<b>TPPL</b>	1	Kecepatan respon (server) pada aplikasi pengembang perangkat lunak mendukung pekerjaan tim anda. (Semakin pentingnya peningkatan waktu respon, maka semakin sangat setuju).
	2	Tim anda menggunakan aplikasi pengembang perangkat lunak dengan optimal dan efisien. (Semakin optimal efisiensi dalam penggunaan, maka semakin sangat setuju).

Kode	No	Pernyataan
	3	Tim anda mudah melakukan instalasi aplikasi pengembangan perangkat lunak yang sedang digunakan. (Semakin mudah instalasinya, maka semakin sangat setuju).
	4	Tim anda mudah dalam menggunakan aplikasi pengembangan perangkat lunak yang sedang digunakan. (Semakin mudah penggunaannya, maka semakin sangat setuju).
	5	Tim anda membutuhkan aplikasi perangkat lunak yang mendukung multi-platform (versi browser, aplikasi mobile, dll atau Windows / OSX / Unix). (Semakin membutuhkan aplikasi multi-platform, maka semakin sangat setuju).
	6	Tim anda dapat dengan mudah mengubah atau menyesuaikan aplikasi pengembang perangkat lunak sesuai keinginan. (Semakin mudah perubahan atau penyesuaian aplikasi, maka semakin sangat setuju).
	7	Tim anda dapat dengan mudah memberi pengamanan pada aplikasi pengembang perangkat lunak yang sedang digunakan. (contoh, memberi password agar pekerjaan tidak dapat diubah dll.) (Semakin mudah aplikasi diberi pengamanan, maka semakin sangat setuju).

Kode	No	Pernyataan
	8	Tim anda membutuhkan pelatihan untuk menguasai aplikasi pengembang perangkat lunak. (Semakin lama waktu pelatihan yang dibutuhkan, maka semakin sangat setuju).
	9	Tim anda termotivasi untuk menggunakan teknologi yang dimiliki aplikasi pengembang perangkat lunak yang sedang digunakan. (Semakin besar motivasi dalam tim, maka semakin sangat setuju).
	10	Tim anda kesulitan dalam memahami bahasa pemrograman yang digunakan pada aplikasi pengembang perangkat lunak. (Semakin sulit bahasa pemrograman, maka semakin sangat setuju).
	11	Tim anda berpengalaman dalam menggunakan aplikasi pengembang perangkat lunak yang sedang digunakan. (Semakin banyak pengalaman dalam menggunakan aplikasi, maka semakin sangat setuju).

Tabel 4.11 Pertanyaan Tentang Komponen Perangkat Lunak

Kode	No	Pernyataan
<b>KPL</b>	1	Tim anda menggunakan kode yang sudah ada pada saat mengembangkan perangkat lunak. (Semakin sering menggunakan kode yang sudah ada, maka semakin sangat setuju, <b>1:</b> Tidak memerlukan pengulangan kode; <b>3:</b> menggunakan kode yang sudah ada dengan wajar; <b>5:</b> kebutuhan yang tinggi dalam menggunakan kode yang sudah ada).



Kode	No	Pernyataan
	2	<p>Penggunaan Third-Party code perlu digunakan tim anda dalam mengembangkan perangkat lunak.</p> <p>(contoh, jika mengembangkan aplikasi menggunakan bahasa PHP dengan Framework, Third-Party code-nya adalah framework libraries; jika mengembangkan aplikasi desktop dengan bahasa C#, Third-Party code-nya adalah .Net classes; jika mengembangkan aplikasi menggunakan bahasa Java, Third-Party code-nya adalah JDK dll.)</p> <p>(Semakin sering menggunakan Third-Party Code, maka semakin sangat setuju,</p> <p><b>1:</b> Tidak diperlukan Third-Party code; <b>3:</b> Third-Party code digunakan hanya di dalam tim; <b>5:</b> Third-Party code digunakan di dalam tim dan pihak ketiga).</p>
	3	<p>Tim anda selalu mengubah kebutuhan saat mengembangkan perangkat lunak.</p> <p>(Semakin sering kebutuhan diubah, maka semakin sangat setuju).</p>

**Tabel 4.12 Pertanyaan Tentang Sources dari Internet**

<b>Kode</b>	<b>No</b>	<b>Pernyataan</b>
<b>SDI</b>	1	<p>Tim anda menggunakan kode yang beredar di internet pada saat mengembangkan perangkat lunak.</p> <p>(Semakin sering menggunakan kode yang beredar di internet, maka semakin sangat setuju <b>1</b>: Tidak memerlukan kode dari internet; <b>3</b>: menggunakan kode yang beredar di internet dengan wajar; <b>5</b>: kebutuhan yang tinggi dalam menggunakan kode yang beredar di internet).</p>
	2	<p>Penggunaan Third-Party code yang berasal dari internet perlu digunakan tim anda pada saat mengembangkan perangkat lunak.</p> <p>(Semakin sering menggunakan Third-Party Code yang berasal dari internet, maka semakin sangat setuju <b>1</b>: Tidak diperlukan Third-Party code; <b>3</b>: Third-Party code yang berasal dari internet digunakan hanya di dalam tim; <b>5</b>: Third-Party code yang berasal dari internet digunakan di dalam tim dan pihak ketiga).</p>

**Tabel 4.13 Pertanyaan Tentang Productivity Factor**

<b>Kode</b>	<b>No</b>	<b>Pernyataan</b>
<b>BP</b>	1	<p>Organisasi anda memiliki tingkat kinerja yang tinggi dan konsisten.</p> <p>(Semakin konsisten tingkat kinerja sebuah organisasi, maka semakin sangat setuju).</p>

Kode	No	Pernyataan
	2	Terdapat komunikasi yang terbuka di dalam organisasi anda. (Semakin terbukanya dalam berkomunikasi, maka semakin sangat setuju).
	3	Anda merasa organisasi memberikan peluang untuk mengasah kemampuan anda (Semakin banyaknya peluang, maka semakin sangat setuju).
	4	Tanggung jawab yang diberikan organisasi kepada anda sudah tepat sasaran. (Semakin tepat sasaran sebuah tanggung jawab, maka semakin sangat setuju).
	5	Organisasi anda adil dalam memberikan gaji atau imbalan. (Semakin tinggi keadilan organisasi, maka semakin sangat setuju).
<b>BT</b>	1	Terdapat keramahan dan kemudahan dalam bersosialisasi di dalam tim anda. (Semakin tinggi tingkat sosialisasi di dalam tim, maka semakin sangat setuju).
	2	Sesama anggota di dalam tim anda saling memberikan identitas yang jelas dengan mudah. (Semakin mudahnya menjelaskan identitas anggota, maka semakin sangat setuju).
	3	Tim anda memiliki kepercayaan diri yang tinggi sehingga perasaan “tim paling unggul” muncul. (Semakin tinggi kepercayaan diri di dalam tim, maka semakin sangat setuju).
	4	Tim anda memiliki sasaran yang tergambar jelas. (Semakin jelas sasaran tim anda, maka semakin sangat setuju).

Kode	No	Pernyataan
	5	Tim anda dapat dengan mudah mengganti anggotanya. (Semakin sering pergantian anggota terjadi, maka semakin sangat setuju).
	6	Stakeholder anda dapat dengan mudah menjalin hubungan yang erat kepada tim anda. (Semakin eratnya hubungan antara stakeholder dan tim, maka semakin sangat setuju).
	7	Komunikasi antar anggota yang terjadi di dalam tim anda mengalir dengan tepat guna. (Semakin efisien anggota tim berkomunikasi, maka semakin sangat setuju).
	8	Tim anda dapat dengan mudah menghasilkan ide-ide yang inovatif. (Semakin banyak ide baru yang dihasilkan, maka semakin sangat setuju).
<b>PDK</b>	1	Anggota tim anda terdiri dari bermacam-macam tipe orang. (Semakin terlihat perbedaan gaya perilaku di dalam tim, maka semakin sangat setuju).
	2	Programmer di dalam tim anda paham dengan segala jenis pemrograman. (Semakin fasih menguasai berbagai bahasa pemrograman, maka semakin sangat setuju).
	3	Tim anda berpengalaman dengan platform pengembangan perangkat yang sedang digunakan, baik perangkat keras maupun lunak. (Semakin mahir menguasai platform, maka semakin sangat setuju).
	4	Manajer proyek pada tim anda sering mengawasi pada saat mengembangkan perangkat lunak. (Semakin tinggi kontrol dari manajer, maka semakin sangat setuju).

Kode	No	Pernyataan
	5	Manajer proyek pada tim anda merasa familiar (menguasai) dengan aplikasi pengembangan perangkat lunak yang sedang digunakan. (Semakin familiar, maka semakin sangat setuju).
<b>LK</b>	1	Lingkungan kerja tim anda mendukung pengembangan perangkat lunak. (Semakin serasi lingkungan kerja anda, maka semakin sangat setuju).
	2	Di lingkungan kerja, tim anda kesulitan dalam memusatkan perhatian pada proyek yang sedang dikerjakan. (Semakin sulit perhatian terpusat, maka semakin sangat setuju).
	3	Anggota tim sering melakukan “Context Switch” pada saat pengembangan perangkat lunak. (Semakin sering Context Switch dilakukan, maka semakin sangat setuju).
	4	Tim anda sering melakukan pergantian susunan anggota pada satu lingkungan kerja atau beberapa lingkungan kerja. (Semakin sering anggota tim berganti posisi, maka semakin sangat setuju).
	5	Lingkungan kerja tim anda mendukung beberapa fasilitas komunikasi seperti tim virtual, atau konferensi video dengan klien dsb. (Semakin lengkap fasilitas komunikasinya, maka semakin sangat setuju).
<b>PPPL</b>	1	Penjadwalan pekerjaan yang terjadi di dalam tim sudah sesuai. (Semakin layak penjadwalan pekerjaan, maka semakin sangat setuju).

Kode	No	Pernyataan
	2	Anggota tim sering mengubah kebutuhan pada saat pengembangan perangkat lunak. (Semakin sering kebutuhan berubah, maka semakin sangat setuju).
	3	Tim anda memiliki jumlah anggota yang banyak. (Semakin banyak anggota tim anda, maka semakin sangat setuju).

#### 4.4 Peninjauan Responden

Target subyek penelitian untuk pengisian kuesioner diarahkan kepada orang yang pernah ikut andil dalam proyek pengembangan perangkat lunak, dimana proyeknya telah diteliti oleh peneliti terdahulu atau belum pernah diteliti sama sekali. Selain itu, kriteria dari subyek penelitian adalah yang termasuk dalam anggota tim pengembang perangkat lunak yang berbasis *Object Oriented Project* (OOP). Target jumlah subyek penelitian adalah 30 orang koresponden, yang masing – masing orang mewakili satu proyek pengembangan perangkat lunak.

Gay dan Diehl berpendapat bahwa, jika penelitiannya bersifat korelasional, maka sampel minimumnya adalah 30 subjek [28].

## BAB V IMPLEMENTASI

Pada bab ini berisi bagaimana penelitian dilakukan, persebaran kuesioner, hambatan dan rintangan dalam pelaksanaan, tanggapan responden hingga data yang dibutuhkan terpenuhi.

### 5.1 Penyebaran Kuesioner

Penyebaran kuesioner dilakukan dengan cara membuat kuesioner *online* dan disebarakan kepada koresponden melalui *email*. Selain itu, penyebaran kuesioner fisik dilakukan secara manual kepada koresponden yang proyeknya dulu menjadi bahan penelitian sebelumnya. Hasil penyebaran kuesioner didapatkan data sebagai berikut:

**Tabel 5.1 Daftar Koresponden**

No	Jenis Kelamin	Jabatan	Nama Proyek Pengembangan PL
1	Laki-laki	Programmer & Designer	Aplikasi Kenaikan Jabatan Fungsional Dosen
2	Laki-laki	Development	Sistem Administrasi Kesiswaan MA
3	Laki-laki	Programmer	Sistem Informasi Kemahasiswaan Dan Alumni
4	Laki-laki	Programmer & Sistem Analis	Sistem Monitoring Tugas Akhir
5	Laki-laki	Mahasiswa	Sistem Pendokumentasian Kegiatan Penelitian
6	Laki-laki	Developer	Aplikasi Mobile School Social Network
7	Laki-laki	Developer	School Social Network
8	Laki-laki	Project Manager	MLM System
9	Laki-laki	Project Manager	Sistem Penjualan PT. Konektindo Koburama
10	Laki-laki	Programmer	Sistem Manajemen Pelatihan (SIMPLE)

<b>No</b>	<b>Jenis Kelamin</b>	<b>Jabatan</b>	<b>Nama Proyek Pengembangan PL</b>
11	Laki-laki	Sistem Analis	E-STNK
13	Laki-laki	Programmer	Sistem Manajemen Tiket Event Online (O-Nick)
14	Perempuan	Project Manager	Sistem Persewaan Tempat/Gedung (PlaceForU)
15	Laki-laki	Project Manager	Mall Search Engine (MSE)
16	Laki-laki	Project Manager	Cookies Land
17	Laki-laki	Sistem Analis	Mandiri Data Dictionary Bank XYZ
18	Perempuan	Technical Writer	Survei & Pelaporan Perbankan
19	Laki-Laki	Programmer	Cash Management System
20	Perempuan	Officer	Syiaexpress
21	Perempuan	FP Perencanaan dan Standarisasi TI	Sistem Informasi Perpajakan
22	Laki-Laki	IT Technician	SMS GATEWAY
23	Laki-Laki	SPV	E-Pallet
24	Laki-Laki	Sistem Analis	Sistem Informasi Akademik Berbasis Android
25	Laki-Laki	Programmer	Aplikasi Meeting Management
26	Laki-Laki	Staff IT	Pengembangan Software ERP
27	Laki-Laki	IT Teknis	Aplikasi Pengukuran Media Meter Panel (GRIDVIS)
28	Laki-Laki	Programmer & Arsitek Software	Pemrograman Workflow untuk Integrasi ERP
29	Laki-Laki	PM	PLN Library Cabang Distribusi Bali
30	Laki-Laki	MT	Migrasi dan Pengembangan Oracle R12

## 5.2 Pengolahan Data.

Berdasarkan kuesioner yang telah terisi, dilakukan



rekap hasil menggunakan *microsoft excel*. Kemudian dilakukan proses pengolahan data sesuai urutan metodologi, yaitu uji asumsi kualitas pengukuran dan analisis inferensial. Uji asumsi kualitas pengukuran menggunakan program SPSS dan analisis inferensial menggunakan *tools SEM GSCA*.

Data lebih lengkapnya dapat dilihat pada Lampiran B.

### **5.3 Hambatan Implementasi**

Hambatan yang terjadi dalam proses pengumpulan data menggunakan kuesioner adalah koresponden yang terkadang susah ditemui. Banyak koresponden yang telah memiliki kesibukan dan tidak punya cukup waktu untuk mengadakan pertemuan perihal pengisian kuesioner. Rata – rata koresponden yang mengalami hal tersebut adalah koresponden yang proyek pengembangannya telah diteliti oleh peneliti sebelumnya.

### **5.4 Solusi Hambatan**

Solusi atas hambatan yang terjadi dalam proses pengumpulan data menggunakan kuesioner adalah dengan terus menindaklanjuti koresponden. Bentuk tindak lanjut yang dilakukan peneliti yaitu, mengirimkan kuesioner via ekspedisi, mencari waktu koresponden yang fleksibel dan mengunggah kuesioner ke internet agar dapat dengan mudah diakses.

*Halaman ini sengaja dikosongkan*

## **BAB VI**

### **HASIL DAN PEMBAHASAN**

Pada bab ini akan dilakukan penjabaran hasil dan pembahasan terhadap hasil yang diperoleh, yang akan digunakan untuk menjawab rumusan masalah yang diangkat dalam penelitian ini.

#### **6.1 Uji Asumsi Kualitas Pengukuran**

Uji Asumsi kualitas pengukuran merupakan pengukuran yang perlu dilakukan sebelum dilakukan analisis inferensial pada *tools* SEM GSCA. Pada uji ini meliputi uji validitas, reliabilitas dan linearitas yang merupakan syarat untuk melakukan pengujian melalui SEM GSCA. Analisis asumsi kualitas ini dilakukan dengan menggunakan *tools* SPSS.

##### **6.1.1 Uji validitas**

Uji validitas dilakukan dengan melakukan Uji Korelasi *Pearson Product Moment*. Dalam uji ini, setiap item akan diuji relasinya dengan skor total variabel yang dimaksud. Dalam hal ini masing-masing item yang ada di dalam variabel X dan Y akan diuji relasinya dengan skor total variabel tersebut. Kemudian hasil tersebut akan dibandingkan dengan *r table*.

Dari hasil survey yang dilakukan, didapatkan 30 koresponden. Sebelum membandingkan dengan *r table*, harus dilakukan nilai DF. Nilai DF didapatkan rumus  $(n-2)$ , jadi *r tabel* yang digunakan untuk perbandingan adalah pada DF 28. Signifikansi yang digunakan pada analisa ini adalah 0,01 pada uji 2 arah (*2 tailed*), jadi nilai *r tabel* yang digunakan perbandingan dengan corrected item total correlation adalah 0,463.

Berikut adalah hasil uji validitas pada indikator Metode Pengembangan Perangkat Lunak (MPPL) ditunjukkan pada Tabel 6.1.

**Tabel 6.1 Uji Validitas MPPL**

<b>Variabel</b>	<b>Pearson Correlation</b>	<b>r tabel</b>	<b>Keterangan</b>
MPPL1	,643**	0,463	Valid
MPPL2	,697**	0,463	Valid
MPPL3	,732**	0,463	Valid
MPPL4	,690**	0,463	Valid
MPPL5	,767**	0,463	Valid
MPPL6	,604**	0,463	Valid
MPPL7	,808**	0,463	Valid
MPPL8	,689**	0,463	Valid

Berdasarkan Tabel 6.1, dapat dilihat nilai korelasi pearson (koefisien validitas) kemudian dibandingkan dengan nilai r tabel dengan rumus  $df = N - 2$  pada  $N=30$  dengan signifikansi 0.01 yang bernilai 0.463. Hal ini berarti bahwa instrument untuk pengukuran variabel MPPL dikatakan valid karena nilai dari ke semua variabel indikator  $\geq 0.463$  sehingga dapat digunakan untuk analisis selanjutnya.

Hasil uji validitas pada indikator Teknologi Pengembangan Perangkat Lunak (TPPL) ditunjukkan pada Tabel 6.2.

**Tabel 6.2 Uji Validitas TPPL**

<b>Variabel</b>	<b>Pearson Correlation</b>	<b>r tabel</b>	<b>Keterangan</b>
TPPL1	,527**	0,463	Valid
TPPL2	,650**	0,463	Valid
TPPL3	,755**	0,463	Valid

Variabel	Pearson Correlation	r tabel	Keterangan
TPPL6	,683**	0,463	Valid
TPPL7	,798**	0,463	Valid
TPPL8	,489**	0,463	Valid
TPPL9	,527**	0,463	Valid
TPPL10	,798**	0,463	Valid
TPPL11	,524**	0,463	Valid

Berdasarkan Tabel 6.2 dapat dilihat nilai korelasi pearson (koefisien validitas) kemudian dibandingkan dengan nilai r tabel dengan rumus  $df = N - 2$  pada  $N=30$  dengan signifikansi 0.01 yang bernilai 0.463. Hal ini berarti bahwa instrument untuk pengukuran variabel TPPL dikatakan valid karena nilai dari ke semua variabel indikator  $\geq 0.463$  sehingga dapat digunakan untuk analisis selanjutnya.

Hasil uji validitas pada indikator Komponen Perangkat Lunak (KPL) ditunjukkan pada Tabel 6.3.

**Tabel 6.3 Uji Validitas KPL**

Variabel	Pearson Correlation	r tabel	Keterangan
KPL1	,896**	0,463	Valid
KPL2	,930**	0,463	Valid
KPL3	,916**	0,463	Valid

Berdasarkan Tabel 6.3 dapat dilihat nilai korelasi pearson (koefisien validitas) kemudian dibandingkan dengan nilai r tabel dengan rumus  $df = N - 2$  pada  $N=30$  dengan signifikansi 0.01 yang bernilai 0.463. Hal ini berarti bahwa instrument untuk pengukuran variabel KPL dikatakan valid karena nilai dari ke semua variabel indikator  $\geq 0.463$  sehingga dapat digunakan untuk analisis selanjutnya.

Hasil uji validitas pada indikator *Sources* Dari Internet (SDI) ditunjukkan pada Tabel 6.4.

**Tabel 6.4 Uji Validitas SDI**

<b>Variabel</b>	<b>Pearson Correlation</b>	<b>r tabel</b>	<b>Keterangan</b>
SDI1	,955**	0,463	Valid
SDI2	,965**	0,463	Valid

Berdasarkan Tabel 6.4 dapat dilihat nilai korelasi pearson (koefisien validitas) kemudian dibandingkan dengan nilai r tabel dengan rumus  $df = N - 2$  pada  $N=30$  dengan signifikansi 0.01 yang bernilai 0.463. Hal ini berarti bahwa instrument untuk pengukuran variabel SDI dikatakan valid karena nilai dari ke semua variabel indikator  $\geq 0.463$  sehingga dapat digunakan untuk analisis selanjutnya.

Hasil uji validitas pada indikator Productivity Factor (PF), yang terdiri dari beberapa item, ditunjukkan pada Tabel 6.5.

**Tabel 6.5 Uji Validitas Productivity Factor**

<b>Variabel</b>	<b>Pearson Correlation</b>	<b>r tabel</b>	<b>Keterangan</b>
BP1	-0,071	0,463	Tidak Valid
BP2	0,209	0,463	Tidak Valid
BP3	-0,019	0,463	Tidak Valid
BP4	,789**	0,463	Valid
BP5	,712**	0,463	Valid
BT1	0,345	0,463	Tidak Valid
BT2	-0,072	0,463	Tidak Valid
BT3	0,33	0,463	Tidak Valid

Variabel	Pearson Correlation	r tabel	Keterangan
BT4	,766**	0,463	Valid
BT5	0,23	0,463	Tidak Valid
BT6	,432*	0,463	Valid
BT7	,724**	0,463	Valid
BT8	,815**	0,463	Valid
PDK1	0,074	0,463	Tidak Valid
PDK2	,717**	0,463	Valid
PDK3	,835**	0,463	Valid
PDK4	,830**	0,463	Valid
PDK5	-0,115	0,463	Tidak Valid
LK1	,816**	0,463	Valid
LK2	0,188	0,463	Tidak Valid
LK3	0,026	0,463	Tidak Valid
LK4	,795**	0,463	Valid
LK5	,791**	0,463	Valid
PPPL1	,855**	0,463	Valid
PPPL2	0,188	0,463	Tidak Valid
PPPL3	,760**	0,463	Valid

Berdasarkan Tabel 6.5 dapat dilihat nilai korelasi pearson (koefisien validitas) kemudian dibandingkan dengan nilai r tabel dengan rumus  $df = N - 2$  pada  $N=30$  dengan signifikansi 0.01 yang bernilai 0.463. Hal ini berarti bahwa beberapa instrument untuk pengukuran *Productivity Factor* dikatakan valid karena nilai dari 13 variabel indikator  $\geq 0.463$  sehingga dapat digunakan untuk analisis selanjutnya.

Adapun variabel BT6 menunjukkan signifikansi, tetapi letak signifikansi berada di 0.05, sehingga tidak dapat digunakan untuk analisis selanjutnya.

### 6.1.2 Uji Reliabilitas

Uji reliabilitas dilakukan dengan menggunakan perbandingan nilai Cronbach's Alpha. Jika nilai alpha  $> 0,7$  artinya reliabilitas mencukupi (*sufficient reliability*). Sementara jika alpha  $> 0,80$  ini mensugestikan seluruh item reliabel dan seluruh tes konsisten secara internal karena memiliki reliabilitas yang kuat [28].

Berikut adalah hasil uji reliabilitas tiap variabel laten.

**Tabel 6.6 Uji Reliabilitas Variabel Laten**

<b>Variabel laten</b>	<b>Cronbach's Alpha</b>	<b>Keterangan</b>
MPPL	0,828	Reliabel
TPPL	0,841	Reliabel
KPL	0,902	Reliabel
SDI	0,911	Reliabel
PF	0,968	Reliabel

Melalui tabel 6.6 dapat disimpulkan bahwa data pada tiap variabel laten bersifat reliabel dan dapat dilakukan proses analisa berikutnya.

### 6.1.3 Uji Linieritas

Pada asumsi uji linearitas ini dilakukan pengujian dengan metode uji linieritas via ANOVA dengan cara melihat nilai F.hitung dan nilai signifikansi di baris *Deviation from Linearity*. Jika nilai F.hitung kurang dari nilai F.tabel dan nilai *Deviation from Linearity* mencapai lebih dari 0,05, maka hubungan antar variabel tersebut linier.

Uji ini dilakukan untuk melihat hubungan antar variabel. Dari Tabel 6.7 diketahui capaian nilai signifikansi antar variabel MPPL dengan PF, MPPL dengan PF, MPPL dengan PF dan MPPL dengan PF.



Tabel 6.7 Hasil Uji Linieritas

Variabel	Nilai F.Hitung	Nilai Sig. Deviation from Linearity	Keterangan
MPPL → PF	0,954	0,529	Linear
TPPL → PF	1,898	0,121	Linear
KPL → PF	1,408	0,256	Linear
SDI → PF	1,605	0,193	Linear

Dari hasil Tabel 6.7 dapat diketahui bahwa semua hubungan antara variabel dependent dengan independent bersifat linear karena nilai F.hitung kurang dari nilai F.tabel (F.tabel: 2,76) dan nilai signifikansi *Deviation from Linearity* lebih dari 0,05.

Asumsi mengenai uji linearitas ini dilakukan karena untuk mengetahui apakah model sebaiknya layak untuk dianalisis lebih lanjut dengan menggunakan metode non linear ataupun linear, seperti analisis yang dapat dilakukan dengan menggunakan metode SEM dan juga mengetahui apakah dua variabel mempunyai hubungan yang linear atau tidak.

## 6.2 Analisis Inferensial

Analisis Inferensial dilakukan dengan menggunakan pendekatan berbasis komponen, yakni dengan alat bantu

*Generalized Structured Component Analysis (GSCA)*. Analisis ini dimulai dari melakukan perancangan model struktural, melakukan perancangan model pengukuran, mengembangkan diagram jalur. Setelah output dari GSCA keluar, dilakukan estimasi terhadap *weight and loading estimate*, *path coefisient estimate*, lalu *means weight*, *loading* dan *path coefficient*.

Berikut adalah langkah-langkah untuk melakukan analisis inferensial dengan alat bantu dengan GSCA:

1. Langkah pertama : Merancang model Struktural (hubungan antar variabel laten)

Perancangan model struktural hubungan antar variabel laten pada GSCA adalah hipotesis penelitian atau rumusan masalah. Pada SEM perancangan model menggunakan teori sedangkan untuk GSCA rujukan untuk merancangan model hubungan antar variabel ada banyak, yakni:

- ✓ Normatif finalitas
- ✓ Teori
- ✓ Hasil penelitian empiris
- ✓ Analogi
- ✓ Normatif non finalitas
- ✓ Rasional

Banyaknya rujukan dalam perancangan model struktural pada GSCA memungkinkan eksplorasi terhadap hubungan antar variabel. Eksplorasi ini membuat rancangan model struktural lebih rasional, sebab tertuang dalam premis-premis.

2. Langkah kedua : Merancang model pengukuran (reflektif atau formatif)

Pada GSCA perancangan model pengukuran merupakan hal yang penting. Kesalahan dalam membuat model pengukuran membuat hasil analisis menjadi bias. Tahap perancangan model pengukuran pada GSCA adalah menentukan apakah indikator dari masing-masing variabel laten bersifat reflektif atau formatif.

Hal ini berbeda dengan SEM, karena model pengukuran pada SEM semua bersifat reflektif sehingga sulit dijelaskan secara detail. Model pengukuran pada SEM merujuk pada definisi operasional variabel saja.

Penentuan sifat indikator apakah bersifat reflektif atau formatif bisa didasarkan pada 4 hal, yakni:

- ✓ Normative Finalitas
- ✓ Teori
- ✓ Penelitian empiris sebelumnya
- ✓ Jika belum ada bisa memakai rasional.

### 3. Langkah ketiga : Mengkonstruksi diagram jalur

Setelah langkah kedua dan ketiga terlaksana, maka pada langkah ketiga dibuat diagram jalur. Fungsi dari diagram jalur ini adalah untuk memudahkan pemahaman tentang model.

### 4. Langkah keempat : Estimasi: *Weight*, *Loading*, dan Koefisien jalur

Langkah keempat ini bisa disebut dengan pendugaan parameter. Proses pendugaan parameter ini digunakan untuk meminimalkan residual dari model yang telah terintegrasi. Metode pendugaan parameter yang biasa digunakan dalam GSCA adalah *Alternating Least Square-ALS*.

Proses perhitungan menggunakan ALS adalah

perhitungan kompleks. Perhitungan adalah *Alternating Least Square-ALS* adalah proses untuk mendapatkan residual yang minimum dilakukan secara iterasi. Iterasi dalam perhitungan ALS akan berhenti ketika kondisi konvergen tercapai.

Pendugaan parameter dalam GSCA meliputi 3 hal, yakni:

- ✓ *Weight* dan *Loading Estimate* : pendugaan ini dilakukan untuk mendapatkan data dari variabel laten. Pendugaan parameter ini biasanya menggunakan pendekatan eigen value dan eigen vektor
- ✓ *Path Coefficient Estimate* : yakni koefisien hubungan antar variabel laten.
- ✓ *Means Weight, Loading, dan Path Coefficient* : yakni pendugaan parameter berupa rata-rata dari subsampel yang akan digunakan untuk *resampling bootstrap*

5. Langkah kelima : *Measures of Fit* (Evaluasi *Goodness of Fit*)

Analisis *Measures of Fit* dalam GSCA akan dilakukan pada 3 model, yakni model struktural, model pengukuran dan model keseluruhan. *Measures of Fit* pada model struktural digunakan untuk mengetahui seberapa besar informasi yang dapat dijelaskan oleh model ( hubungan antar variabel laten). *Measures of Fit* pada model pengukuran digunakan untuk menguji apakah instrumen yang digunakan dalam penelitian ini valid dan reliabel atau tidak. *Measures of Fit* pada model keseluruhan adalah untuk melihat apakah model sudah sesuai atau belum.

- ✓ *Measures of Fit* untuk model struktural

Measures of Fit untuk model struktural diukur dengan FIT. FIT merupakan varian total dari semua variabel yang dapat dijelaskan dalam model struktural. Nilai FIT berada dalam rentang 0 sampai 1. Semakin besar nilai FIT maka semakin besar proporsi variabel yang dapat dijelaskan oleh model struktural.

Selain FIT ada juga *adjusted FIT* (AFIT). AFIT digunakan untuk perbandingan model, semakin besar nilai AFIT sebuah model maka semakin baik model.

✓ *Measures of Fit* untuk model pengukuran

Apabila indikator yang digunakan dalam penelitian bersifat reflektif maka diperlukan evaluasi berupa kalibrasi instrumen. Kalibrasi instrumen ini dilakukan untuk memastikan bahwa instrumen yang digunakan dalam penelitian valid dan reliabel.

> *Convergent Validity*

Validitas ini dilihat berdasarkan korelasi antara skor indikator reflektif dengan skor variabel latennya. Jika loading berada dalam rentang 0.5 hingga 0.6 maka instrumen dianggap valid. Selain itu validitas dapat dilihat juga dari hasil pengujian, jika hasil signifikan ( $p < 0.05$ ) maka bisa dikatakan valid.

> *Discriminant Validity*

Validitas ini pada indikator reflektif dinilai berdasarkan nilai (*Average Variance Extracted*) AVE. cara yang digunakan adalah membandingkan *Square Root* dari *Average Variance Extracted* dari setiap variabel laten dengan korelasi antar variabel lainnya di dalam model. Jika nilai *Square Root* dari *Average*

*Variance Explained* lebih besar dari korelasi dengan seluruh variabel laten lainnya, maka dikatakan instrumen memiliki *Discriminant Validity* yang baik.

> *Internal Consistency Reliability*

Pengukuran reliabilitas sebuah variabel dilakukan dengan melihat nilai *alpha*, jika nilai  $\alpha \geq 0.6$  maka instrumen reliabel.

> *Substantive Content*

Untuk indikator formatif, pengukuran validitas dilihat dari nilai *Substantive Content*. Caranya dengan melihat apakah *weight* signifikan atau tidak. Jika nilai *weight* ( $p < 0.05$ ) signifikan berarti valid.

✓ *Measures of Fit* untuk model keseluruhan

*Measures of Fit* untuk model keseluruhan adalah pengukuran yang dilakukan terhadap model struktural dan model pengukuran yang sudah diintegrasikan menjadi model keseluruhan. Berikut adalah tabel pengukuran *Goodness of Fit* dari model keseluruhan.

**Tabel 6.8 Goodness of Fit dari Model Keseluruhan**

No	<i>Goodness of Fit</i>	Cut-off
1	SRMR	$\leq 0.08$
2	GFI	$\geq 0.90$

**Tabel 6.9 Kriteria SRMR**

No	SRMR	Keterangan
1	$< 0,05$	Close fit (model sangat sesuai)
2	$0,05 - 0,08$	Good fit (model sesuai)

No	SRMR	Keterangan
3	0,08 - 0,1	Marginal fit (model cukup sesuai)
4	> 0,1	Poor fit (model tidak sesuai)

6. Langkah keenam : Pengujian hipotesis ini menggunakan output dari GSCA dengan melihat *path coefficient*.

Jika nilai pada *path coefficient* terdapat tanda bintang (\*) maka hipotesis diterima. Apabila nilai *path coefficient* tidak terdapat tanda bintang maka hipotesis ditolak.

Data yang diperoleh dari penyebaran kuesioner kemudian dimasukkan dalam model penelitian pada aplikasi *online* GeSCA. Hasil lengkap dari perhitungan yang dilakukan aplikasi *online* GeSCA dapat dilihat pada Lampiran C.

### 6.2.1 Identifikasi Goodness of FIT

Tabel 6.10 Model Fit

Model Fit	
<b>FIT</b>	0.557
<b>AFIT</b>	0.523
<b>GFI</b>	0.976
<b>SRMR</b>	0.216
<b>NPAR</b>	76

**FIT = 0,557**

FIT menunjukkan varian total dari semua variabel yang dapat dijelaskan oleh model tertentu. Nilai FIT berkisar dari angka 0 sampai 1. Berdasarkan Tabel 6.10 diketahui bahwa model yang terbentuk dapat menjelaskan semua variabel yang ada sebesar 0.557.

Metode pengembangan perangkat lunak, teknologi pengembangan perangkat lunak, komponen perangkat lunak, *sources* dari internet, dan *productivity factor* dapat dijelaskan oleh model sebesar 55,7 % dan sisanya 44,3 % dapat dijelaskan oleh variabel lain. Berarti model cukup baik untuk menjelaskan fenomena yang dikaji.

### **AFIT = 0,523**

Di dalam model ini terdapat multi variabel yang mempengaruhi *productivity factor*, oleh karena itu sebaiknya melakukan interpretasi ketepatan model dengan FIT yang sudah terkoreksi *Adjusted FIT* (AFIT).

Jika dilihat dari nilai AFIT, keragaman metode pengembangan perangkat lunak, teknologi pengembangan perangkat lunak, komponen perangkat lunak, *sources* dari internet, dan *productivity factor* yang dapat dijelaskan oleh model adalah sebesar 52,3 % dan sisanya 47,7 % dapat dijelaskan oleh variabel lain.

### **GFI = 0,976**

*Unweighted Least Square* (GFI) ini menunjukkan bahwa model sudah sesuai atau belum, dengan rentang nilai 0 hingga 1.

Pada penelitian ini, nilai GFI yang dihasilkan sebesar 0,976. Dengan nilai sebesar 0,976 berarti bahwa model keseluruhan sudah sangat sesuai, sebab nilainya mendekati 1.

### **SRMR = 0,216**

*Standardize Root Mean Square Residual* (SRMR) merupakan penambahan ukuran model fit. Jika nilai SRMR mendekati 0 maka hal ini mengindikasikan kesesuaian model keseluruhan.

Pada penelitian ini nilai SRMR model sebesar 0,216 berarti model berdasarkan SRMR tidak sesuai. Sedangkan



menurut Imam Ghozali, nilai SRMR < 0.08 [20]. Hal ini mungkin disebabkan oleh jumlah sampel yang sedikit, sehingga variasi nilai yang dimiliki kurang baik. Selain itu juga disebabkan oleh proyek yang diteliti ada yang berskala kecil sehingga menyebabkan jawaban yang dihasilkan berskala kecil juga.

### **NPAR = 76**

*Number of Free Parameters Eestimated* (NPAR) menunjukkan banyaknya parameter bebas yang digunakan dalam perhitungan alat bantu GSCA, termasuk *weights, loadings, and path coefficient*.

### **6.2.2 Identifikasi R Square**

*Square* digunakan untuk mencari besarnya pengaruh variabel independen terhadap variabel dependen secara simultan atau bersama-sama. Arti dari *R Square* tersebut apabila nilai *R Square* mendekati 1, maka secara bersama-sama variabel independen berpengaruh kuat terhadap variabel dependen dan apabila *R square* mendekati angka nol, maka variabel independen tidak berpengaruh terhadap variabel dependen.

**Tabel 6.11 Identifikasi R Square**

<b>R square of Latent Variable</b>	
<b>ProdFactor</b>	0.808
<b>MetodePPL</b>	0
<b>TeknologiPPL</b>	0
<b>KomponenPL</b>	0
<b>SDInternet</b>	0

Pada Tabel 6.15 dapat dilihat bahwa nilai *R Square* pada *productivity factor* adalah 0,808. Hal ini menunjukkan ketika terjadi peningkatan terhadap metode pengembangan

perangkat lunak, teknologi pengembangan perangkat lunak, komponen perangkat lunak, dan *sources* dari internet, maka dapat mempengaruhi *productivity factor* sebesar 81%, sedangkan sisanya merupakan pengaruh dari variabel lain yang tidak masuk dalam model.

## 6.2.3 Identifikasi Indikator-Indikator

### 6.2.3.1 MPPL

**Tabel 6.12 Identifikasi Indikator MPPL**

Variable	Loading		
	Estimate	SE	CR
<b>Metode</b>	<b>AVE = 0,509, Alpha = 0,828</b>		
<b>MPPL1</b>	0.657	0.169	3.88 <sup>*</sup>
<b>MPPL2</b>	0.711	0.102	7.0 <sup>*</sup>
<b>MPPL3</b>	0.758	0.108	7.02 <sup>*</sup>
<b>MPPL4</b>	0.760	0.104	7.29 <sup>*</sup>
<b>MPPL5</b>	0.808	0.047	17.11 <sup>*</sup>
<b>MPPL6</b>	0.517	0.156	3.32 <sup>*</sup>
<b>MPPL7</b>	0.837	0.055	15.2 <sup>*</sup>
<b>MPPL8</b>	0.604	0.150	4.01 <sup>*</sup>

Dari hasil yang diperoleh ditunjukkan bahwa semua indikator pada variabel MPPL memiliki pengaruh yang signifikan terhadap variabel MPPL. Hal tersebut dilihat dari titik nilai krisis (CR) yang lebih besar dari 1,96 ( $CR > 1,96$ ) dan terdapat indikator (\*) pada nilai yang dimunculkan.

Dari semua indikator pada Metode Pengembangan Perangkat Lunak, indikator MPPL 5 yang paling menggambarkan dari variabel MPPL yaitu tentang Tingkat motivasi pengembang dalam menggunakan berbagai metode pengembangan perangkat lunak.

Nilai AVE (*average variance extracted*) menunjukkan tingkat konvergensi yang dimiliki oleh variabel. Konvergensi ini adalah kerapatan koordinat-koordinat. Pada variabel MPPL, nilai AVE sama dengan 0.5, sehingga dikatakan variabel MPPL memiliki kemampuan yang cukup dalam mewakili skor data asli.

Selain itu, nilai AVE juga diperlukan dalam membandingkan nilai akar AVE dengan nilai korelasi konstruk pada variabel lain *Correlations of Latent Variables (SE)*. Nilai AVE 0,509 setelah diakar kuadratkan menjadi 0,713. Sedangkan, nilai variabel laten yang lain bernilai - 0,051 pada PF, 0,679 pada TPPL, 0,290 pada KPL dan - 0,064 pada SDI. Apabila nilai akar kuadrat dari AVE bernilai di bawah nilai-nilai tersebut, diasumsikan persebaran data pada MPPL masih kurang maksimal pada nilai yang besar. Sehingga, nilai *discriminant validity* masih berada di bawah nilai perbandingan variabel laten lain.

### 6.2.3.2 TPPL

**Tabel 6.13 Identifikasi Indikator TPPL**

Variable	Loading		
	Estimate	SE	CR
<b>Teknologi</b>	<b>AVE = 0,416, Alpha = 0,759</b>		
<b>TPPL1</b>	0.392	0.160	2.46 <sup>*</sup>
<b>TPPL2</b>	0.763	0.082	9.26 <sup>*</sup>
<b>TPPL3</b>	0.901	0.041	22.24 <sup>*</sup>
<b>TPPL4</b>	0.796	0.065	12.22 <sup>*</sup>
<b>TPPL5</b>	0.593	0.142	4.16 <sup>*</sup>
<b>TPPL6</b>	0.812	0.064	12.69 <sup>*</sup>
<b>TPPL7</b>	0.909	0.037	24.57 <sup>*</sup>

Variable	Estimate	SE	CR
<b>TPPL8</b>	0.133	0.251	0.53
<b>TPPL10</b>	-0.063	0.284	0.22
<b>TPPL11</b>	0.345	0.195	1.77

Dari hasil yang diperoleh ditunjukkan bahwa hampir semua indikator pada variabel TPPL memiliki pengaruh yang signifikan terhadap variabel TPPL. Hal tersebut dilihat dari titik nilai kritis (CR) yang lebih besar dari 1,96 ( $CR > 1,96$ ) dan terdapat indikator (\*) pada nilai yang dimunculkan.

Dari semua indikator pada Teknologi Pengembangan Perangkat Lunak, indikator TPPL 7 yang paling menggambarkan dari variabel TPPL yaitu tentang Tingkat keamanan pada aplikasi pengembangan perangkat lunak.

Nilai AVE (*average variance extracted*) menunjukkan tingkat konvergensi yang dimiliki oleh variabel. Konvergensi ini adalah kerapatan koordinat-koordinat. Pada variabel TPPL, nilai AVE kurang dari 0.5, sehingga dikatakan variabel TPPL memiliki kemampuan yang kurang dalam mewakili skor data asli.

Selain itu, nilai AVE juga diperlukan dalam membandingkan nilai akar AVE dengan nilai korelasi konstruk pada variabel lain *Correlations of Latent Variables (SE)*. Nilai AVE 0,416 setelah diakar kuadratkan menjadi 0,645. Sedangkan, nilai variabel laten yang lain bernilai 0,020 pada PF, 0,679 pada MPPL, 0,069 pada KPL dan -0,024 pada SDI. Apabila nilai akar kuadrat dari AVE bernilai di bawah nilai-nilai tersebut, diasumsikan persebaran data pada TPPL masih kurang maksimal pada nilai yang besar. Sehingga, nilai *discriminant validity* masih berada di bawah nilai perbandingan variabel laten lain.

### 6.2.3.3 KPL

**Tabel 6.14 Identifikasi Indikator KPL**

Variable	Loading		
	Estimate	SE	CR
<b>Komponen</b>	<b>AVE = 0,836, Alpha = 0,902</b>		
<b>KPL1</b>	-0.890	0.045	20.0*
<b>KPL2</b>	-0.933	0.024	38.39*
<b>KPL3</b>	-0.919	0.030	30.96*

Dari hasil yang diperoleh ditunjukkan bahwa semua indikator pada variabel KPL memiliki pengaruh yang signifikan terhadap variabel KPL. Hal tersebut dilihat dari titik nilai krisis (CR) yang lebih besar dari 1,96 ( $CR > 1,96$ ) dan terdapat indikator (\*) pada nilai yang dimunculkan.

Dari semua indikator pada Komponen Perangkat Lunak, indikator KPL 2 yang paling menggambarkan dari variabel KPL yaitu tentang Tingkat penggunaan Third-Party code dalam mengembangkan perangkat lunak.

Nilai AVE (*average variance extracted*) menunjukkan tingkat konvergensi yang dimiliki oleh variabel. Konvergensi ini adalah kerapatan koordinat-koordinat. Pada variabel KPL, nilai AVE lebih dari 0.5, sehingga dikatakan variabel KPL memiliki kemampuan yang baik dalam mewakili skor data asli.

Selain itu, nilai AVE juga diperlukan dalam membandingkan nilai akar AVE dengan nilai korelasi konstruk pada variabel lain *Correlations of Latent Variables (SE)*. Nilai AVE 0,836 setelah diakar kuadratkan menjadi 0,914. Sedangkan, nilai variabel laten yang lain bernilai -0,820 pada PF, 0,069 pada TPPL, 0,290 pada MPPL dan -0,817 pada SDI. Apabila nilai akar kuadrat dari AVE bernilai di bawah nilai-nilai tersebut, diasumsikan

persebaran data pada KPL masih kurang maksimal pada nilai yang besar. Sehingga, nilai *discriminant validity* masih berada di bawah nilai perbandingan variabel laten lain.

#### 6.2.3.4 SDI

**Tabel 6.15 Identifikasi Indikator SDI**

Variable	Loading		
	Estimate	SE	CR
<b>Internet</b>	<b>AVE = 0,921, Alpha = 0,911</b>		
<b>SDI1</b>	0.963	0.020	48.89 <sup>*</sup>
<b>SDI2</b>	0.957	0.024	40.61 <sup>*</sup>

Dari hasil yang diperoleh ditunjukkan bahwa semua indikator pada variabel SDI memiliki pengaruh yang signifikan terhadap variabel SDI. Hal tersebut dilihat dari titik nilai krisis (CR) yang lebih besar dari 1,96 ( $CR > 1,96$ ) dan terdapat indikator (\*) pada nilai yang dimunculkan.

Dari semua indikator pada *Sources* Dari Internet, indikator SDI 1 yang paling menggambarkan dari variabel SDI yaitu tentang Tingkat penggunaan ulang kode yang berasal dari internet dalam mengembangkan perangkat lunak.

Nilai AVE (*average variance extracted*) menunjukkan tingkat konvergensi yang dimiliki oleh variabel. Konvergensi ini adalah kerapatan koordinat-koordinat. Pada variabel SDI, nilai AVE lebih dari 0.5, sehingga dikatakan variabel SDI memiliki kemampuan yang baik dalam mewakili skor data asli.

Selain itu, nilai AVE juga diperlukan dalam membandingkan nilai akar AVE dengan nilai korelasi konstruk pada variabel lain *Correlations of Latent Variables* (SE). Nilai AVE 0,921 setelah diakar kuadratkan menjadi 0,959. Sedangkan, nilai variabel laten yang lain bernilai -

0,875 pada PF, 0,024 pada TPPL, -0,817 pada KPL dan -0,064 pada MPPL. Apabila nilai akar kuadrat dari AVE bernilai di bawah nilai-nilai tersebut, diasumsikan persebaran data pada SDI masih kurang maksimal pada nilai yang besar. Sehingga, nilai *discriminant validity* masih berada di bawah nilai perbandingan variabel laten lain.

### 6.2.3.5 PF

**Tabel 6.16 Identifikasi Indikator PF**

Variable	Loading		
	Estimate	SE	CR
<b>ProdFactor</b>	<b>AVE = 0,725, Alpha = 0,968</b>		
<b>BP4</b>	0.834	0.038	21.91 <sup>*</sup>
<b>BP5</b>	0.836	0.058	14.48 <sup>*</sup>
<b>BT4</b>	0.877	0.055	15.95 <sup>*</sup>
<b>BT7</b>	0.857	0.087	9.84 <sup>*</sup>
<b>BT8</b>	0.845	0.042	20.01 <sup>*</sup>
<b>PDK2</b>	0.794	0.063	12.51 <sup>*</sup>
<b>PDK3</b>	0.857	0.071	12.05 <sup>*</sup>
<b>PDK4</b>	0.875	0.060	14.58 <sup>*</sup>
<b>LK1</b>	0.838	0.055	15.32 <sup>*</sup>
<b>LK4</b>	0.876	0.051	17.32 <sup>*</sup>
<b>LK5</b>	0.853	0.055	15.64 <sup>*</sup>
<b>PPPL1</b>	0.875	0.066	13.32 <sup>*</sup>
<b>PPPL3</b>	0.849	0.065	13.02 <sup>*</sup>

Dari hasil yang diperoleh pada Tabel 6.16, ditunjukkan bahwa semua indikator pada variabel PF memiliki pengaruh

yang signifikan terhadap variabel PF. Hal tersebut dilihat dari titik nilai krisis (CR) yang lebih besar dari 1,96 ( $CR > 1,96$ ) dan terdapat indikator (\*) pada nilai yang dimunculkan.

Dari semua indikator pada *Productivity Factor*, indikator BP 4 yang paling menggambarkan dari variabel PF yaitu tentang Tingkat peluang dan tanggung jawab yang diberikan organisasi.

Nilai AVE (*average variance extracted*) menunjukkan tingkat konvergensi yang dimiliki oleh variabel. Konvergensi ini adalah kerapatan koordinat-koordinat. Pada variabel PF, nilai AVE lebih dari 0.5, sehingga dikatakan variabel PF memiliki kemampuan yang baik dalam mewakili skor data asli.

Selain itu, nilai AVE juga diperlukan dalam membandingkan nilai akar AVE dengan nilai korelasi konstruk pada variabel lain *Correlations of Latent Variables (SE)*. Nilai AVE 0,725 setelah diakar kuadratkan menjadi 0,851. Sedangkan, nilai variabel laten yang lain bernilai -0,051 pada MPPL, 0,020 pada TPPL, -0,820 pada KPL dan 0,875 pada SDI. Apabila nilai akar kuadrat dari AVE bernilai di bawah nilai-nilai tersebut, diasumsikan persebaran data pada PF masih kurang maksimal pada nilai yang besar. Sehingga, nilai *discriminant validity* masih berada di bawah nilai perbandingan variabel laten lain.

### 6.3 Pengujian Hipotesis

Pada tahap ini akan dilakukan uji hipotesis yang berdasarkan hasil analisis koefisien jalur menggunakan perangkat GeSCA. Uji hipotesis ini bertujuan untuk membuktikan kebenaran hipotesis yang telah dirumuskan pada tahap perumusan hipotesis. Berikut ini adalah hipotesis yang telah dirumuskan:

**H1:** Metode Pengembangan Perangkat Lunak berpengaruh pada *Productivity Factor*, sehingga Nilai ER / PF mengalami penurunan.



**H2:** Teknologi Pengembangan Perangkat Lunak berpengaruh pada *Productivity Factor*, sehingga Nilai ER / PF mengalami penurunan.

**H3:** Komponen Perangkat Lunak berpengaruh pada *Productivity Factor*, sehingga Nilai ER / PF mengalami penurunan.

**H4:** Sumber dari internet berpengaruh pada *Productivity Factor*, sehingga Nilai ER / PF mengalami penurunan.

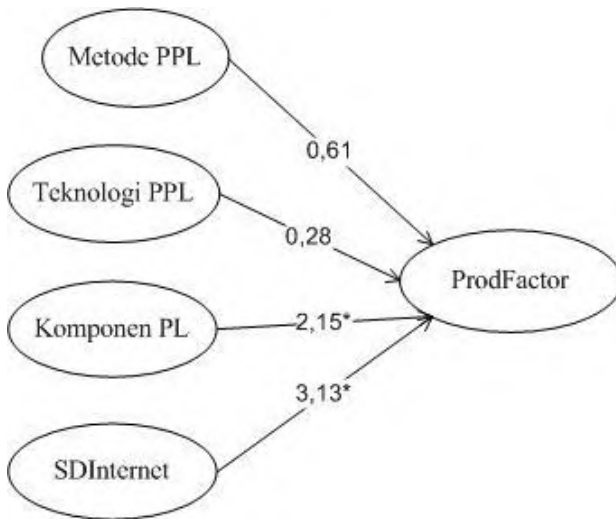
Pada Tabel 6.12 akan disajikan hasil dari penghitungan koefisien jalur pada GeSCA yang menunjukkan pengaruh antar variabel laten.

**Tabel 6.17 Path Coefficients**

<b>Path Coefficients</b>			
	<b>Estimate</b>	<b>SE</b>	<b>CR</b>
<b>Metode-&gt;ProdFactor</b>	0.143	0.234	0.61
<b>Teknologi-&gt;ProdFactor</b>	-0.063	0.224	0.28
<b>Komponen-&gt;ProdFactor</b>	-0.401	0.187	2.15*
<b>Internet-&gt;ProdFactor</b>	0.558	0.178	3.13*

CR\* = significant at .05 level

Pada Gambar 6.1 Hasil Uji Hipotesis merupakan gambaran dari nilai CR atau besar nilai pengaruh antar variabel laten yang digambarkan melalui konseptual model.



**Gambar 6.1 Hasil Uji Hipotesis**

Berikut ini merupakan penjelasan hasil uji hipotesis dari hasil ilustrasi Tabel 6.12 Path Coefficients dan Gambar 6.1:

**1. H1: Metode Pengembangan Perangkat Lunak berpengaruh pada *Productivity Factor*, sehingga Nilai ER / PF mengalami penurunan.**

Pada Tabel 6.12 Path Coefficients dapat dilihat bahwa Metode Pengembangan berpengaruh positif terhadap *Productivity Factor* dengan nilai 0,143(0,61) yang tidak signifikan pada 0,05 menandakan pengaruh yang lemah.

Tidak ada tanda bintang (\*) pada nilai critical ratios menunjukkan pengaruhnya tidak signifikan, karena batasan keterkaitan variabel dinyatakan signifikan apabila nilai  $CR > 1,96$ . Terkait hasil analisis penelitian ini menunjukkan dengan kata lain, semakin tinggi penggunaan Metode Pengembangan Perangkat Lunak,

maka semakin tinggi pula *Productivity Factor* dengan pengaruh yang lemah. Artinya hipotesis 1 diterima.

**2. H2: Teknologi Pengembangan Perangkat Lunak berpengaruh pada *Productivity Factor*, sehingga Nilai ER / PF mengalami penurunan.**

Pada Tabel 6.12 Path Coefficients dapat dilihat bahwa Teknologi Pengembangan berpengaruh negatif terhadap *Productivity Factor* dengan nilai Estimate di -0,063 dan nilai CR 0,28 yang tidak signifikan pada 0,05 menandakan pengaruh yang lemah.

Tidak ada tanda bintang (\*) pada nilai critical ratios menunjukkan pengaruhnya tidak signifikan, karena batasan keterkaitan variabel dinyatakan signifikan apabila nilai  $CR > 1,96$ . Terkait hasil analisis penelitian ini menunjukkan dengan kata lain, semakin tinggi penggunaan Teknologi Pengembangan Perangkat Lunak, maka semakin rendah *Productivity Factor* dengan pengaruh yang lemah. Artinya hipotesis 2 ditolak.

**3. H3: Komponen Perangkat Lunak berpengaruh pada *Productivity Factor*, sehingga Nilai ER / PF mengalami penurunan.**

Pada Tabel 6.12 Path Coefficients dapat dilihat bahwa Komponen PL berpengaruh negatif terhadap *Productivity Factor* dengan nilai Estimate di -0,401 dan nilai CR 2,15 yang signifikan pada 0,05 menandakan pengaruh yang kuat.

Tanda bintang (\*) pada nilai critical ratios menunjukkan bahwa Komponen Perangkat Lunak berpengaruh signifikan terhadap *Productivity Factor*. Dengan kata lain, semakin tinggi penggunaan Komponen Perangkat Lunak, maka semakin rendah *Productivity Factor* dengan pengaruh yang kuat. Artinya hipotesis 3 diterima.

**4. H4: Sumber dari internet berpengaruh pada *Productivity Factor*, sehingga Nilai ER / PF mengalami penurunan.**

Pada Tabel 6.12 Path Coefficients dapat dilihat bahwa Sumber yang berasal dari Internet berpengaruh positif terhadap *Productivity Factor* dengan nilai Estimate di 0,558 dan nilai CR 3,13 yang signifikan pada 0,05 menandakan pengaruh yang kuat.

Tanda bintang (\*) pada nilai critical ratios menunjukkan bahwa Sumber yang berasal dari Internet berpengaruh signifikan terhadap *Productivity Factor*. Dengan kata lain, semakin tinggi penggunaan alat bantu yang bersumber dari Internet, maka semakin tinggi pula *Productivity Factor* dengan pengaruh yang kuat. Artinya hipotesis 4 ditolak.

Tabel 6.13 di bawah merupakan rangkuman pengujian hipotesis yang telah dilakukan.

**Tabel 6.18 Hasil Hipotesis**

<b>Hipotesis</b>		<b>Hasil</b>
H1	<b>Metode Pengembangan Perangkat Lunak berpengaruh pada <i>Productivity Factor</i>, sehingga Nilai ER / PF mengalami penurunan</b>	Diterima
H2	<b>Teknologi Pengembangan Perangkat Lunak berpengaruh pada <i>Productivity Factor</i>, sehingga Nilai ER / PF mengalami penurunan</b>	Ditolak
H3	<b>Komponen Perangkat Lunak berpengaruh pada <i>Productivity Factor</i>, sehingga ER / PF mengalami penurunan</b>	Diterima
H4	<b>Sumber dari internet berpengaruh pada <i>Productivity Factor</i>, sehingga ER / PF mengalami penurunan</b>	Ditolak

## 6.4 Interpretasi Hasil Uji Hipotesis

Berikut ini merupakan interpretasi dari hasil uji hipotesis yang telah dilakukan pada penelitian.

1. H1: Metode Pengembangan Perangkat Lunak berpengaruh pada *Productivity Factor*, sehingga Nilai ER / PF mengalami penurunan.

Dari analisis menggunakan aplikasi GeSCA, didapatkan hasil bahwa Metode Pengembangan Perangkat Lunak berpengaruh positif terhadap *Productivity Factor* dengan pengaruh yang lemah. Nilai *estimate* koefisien jalur yang mencapai angka 0.143, nilai SE 0.234, dan nilai CR sebesar 0,61.

Range nilai *estimate* berkisar antara negatif satu (-1) sampai satu (1), dengan nilai *estimate* sebesar 0,143 dapat diartikan bahwa, semakin tinggi penggunaan Metode Pengembangan Perangkat Lunak, maka semakin tinggi pula nilai *Productivity Factor* yang dihasilkan pada saat *software development*.

Nilai CR yang didapatkan sebesar 0,61, tidak ada tanda bintang setelah besaran angka CR menunjukkan bahwa, Metode Pengembangan Perangkat Lunak berpengaruh lemah terhadap naiknya nilai *Productivity Factor* pada metode UCP. Hal ini berarti jika tim *software developer* menggunakan metode pengembangan yang modern, maka pengaruhnya lemah terhadap peningkatan nilai *Productivity Factor*.

Nilai yang tidak signifikan pada pengaruh variabel Metode Pengembangan Perangkat Lunak terhadap *Productivity Factor* disebabkan oleh sangat minimnya korelasi antara variabel Metode Pengembangan Perangkat Lunak terhadap peningkatan *Productivity Factor*. Hal ini mengindikasikan apabila sebuah tim *software developer* menerapkan Metode Pengembangan

Perangkat Lunak dengan efektif, maka belum tentu nilai *Productivity Factor* yang dihasilkan akan besar.

2. H2: Teknologi Pengembangan Perangkat Lunak berpengaruh pada *Productivity Factor*, sehingga Nilai ER / PF mengalami penurunan.

Dari analisis menggunakan aplikasi GeSCA, didapatkan hasil bahwa Teknologi Pengembangan Perangkat Lunak berpengaruh negatif terhadap *Productivity Factor* dengan pengaruh yang lemah. Nilai *estimate* koefisien jalur yang mencapai angka -0.063, nilai SE 0.224, dan nilai CR sebesar 0,28.

Range nilai *estimate* berkisar antara negatif satu (-1) sampai satu (1), dengan nilai *estimate* sebesar -0.063 dapat diartikan bahwa, semakin tinggi penggunaan Teknologi Pengembangan Perangkat Lunak, maka semakin rendah nilai *Productivity Factor* yang dihasilkan pada saat *software development*.

Nilai CR yang didapatkan sebesar 0,28, tidak ada tanda bintang setelah besaran angka CR menunjukkan bahwa, Teknologi Pengembangan Perangkat Lunak berpengaruh lemah terhadap turunnya nilai *Productivity Factor* pada metode UCP. Hal ini berarti jika tim *software developer* menggunakan aplikasi yang canggih, maka pengaruhnya lemah terhadap penurunan nilai *Productivity Factor*.

Nilai yang tidak signifikan pada pengaruh variabel Teknologi Pengembangan Perangkat Lunak terhadap *Productivity Factor* disebabkan oleh sangat minimnya korelasi antara variabel Teknologi Pengembangan Perangkat Lunak terhadap penurunan *Productivity Factor*. Hal ini mengindikasikan apabila sebuah tim *software developer* menggunakan aplikasi Pengembangan Perangkat Lunak dengan efektif, maka

belum tentu nilai *Productivity Factor* yang dihasilkan akan kecil.

3. H3: Komponen Perangkat Lunak berpengaruh pada *Productivity Factor*, sehingga Nilai ER / PF mengalami penurunan.

Dari analisis menggunakan aplikasi GeSCA, didapatkan hasil bahwa Komponen Perangkat Lunak berpengaruh negatif terhadap *Productivity Factor* dengan pengaruh yang kuat. Nilai *estimate* koefisien jalur yang mencapai angka -0.401, nilai SE 0.187, dan nilai CR sebesar 2,15\*.

Range nilai *estimate* berkisar antara negatif satu (-1) sampai satu, dengan nilai *estimate* sebesar -0,401 dapat diartikan bahwa, semakin tinggi penggunaan Komponen Perangkat Lunak, maka semakin rendah nilai *Productivity Factor* yang dihasilkan pada saat *software development*.

Nilai CR yang didapatkan sebesar 2,15\*, tanda bintang setelah besaran angka CR menunjukkan bahwa Komponen Perangkat Lunak berpengaruh kuat terhadap penurunan nilai *Productivity Factor* pada metode UCP. Hal ini berarti jika tim *software developer* menggunakan segala komponen yang ada pada *tools*, maka pengaruhnya kuat terhadap penurunan nilai *Productivity Factor*.

Nilai signifikansi pada pengaruh variabel Komponen Perangkat Lunak terhadap variabel *Productivity Factor* disebabkan oleh banyaknya korelasi antara variabel Komponen Perangkat Lunak terhadap penurunan *Productivity Factor*. Hal ini mengindikasikan apabila sebuah tim *software developer* menggunakan Komponen Perangkat Lunak dengan efektif, akan berdampak langsung pada turunnya nilai *Productivity Factor*.

4. H4: Sumber dari internet berpengaruh pada *Productivity Factor*, sehingga ER / PF mengalami penurunan.

Dari analisis menggunakan aplikasi GeSCA, didapatkan hasil bahwa *tools* yang bersumber dari Internet berpengaruh positif terhadap *Productivity Factor* dengan pengaruh yang kuat. Nilai *estimate* koefisien jalur yang mencapai angka 0.558, nilai SE 0.178, dan nilai CR sebesar 3,13\*.

Range nilai *estimate* berkisar antara negatif satu (-1) sampai satu, dengan nilai *estimate* sebesar 0.558 dapat diartikan bahwa, semakin tinggi penggunaan bantuan yang berasal dari Internet, maka semakin tinggi pula nilai *Productivity Factor* yang dihasilkan pada saat *software development*..

Nilai CR yang didapatkan sebesar 3,13\*, tanda bintang setelah besaran angka CR menunjukkan bahwa *tools* yang bersumber dari Internet berpengaruh kuat terhadap peningkatan nilai *Productivity Factor* pada metode UCP. Hal ini berarti jika tim *software developer* mendapatkan inspirasi dari Internet, maka pengaruhnya kuat terhadap peningkatan nilai *Productivity Factor*.

Nilai signifikansi pada pengaruh variabel *Sources* atau Sumber Dari Internet terhadap variabel *Productivity Factor* adalah yang paling tinggi dibanding dari pengaruh antar variabel yang lain, hal ini disebabkan oleh korelasi yang sangat signifikan antara variabel *Sources* atau Sumber Dari Internet terhadap peningkatan *Productivity Factor*. Hal ini mengindikasikan apabila sebuah tim *software developer* menggunakan *tools* atau *code* yang didapat dari Internet, akan berdampak langsung pada meningkatnya nilai *Productivity Factor*.



## LAMPIRAN A

### ANGKET (KUESIONER)

#### **Peruntukan: Developer (Pengembang Perangkat Lunak)**

#### **KUESIONER**

Dengan hormat,

bersama ini saya mohon kesediaan Saudara/i untuk mengisi kuesioner yang telah saya siapkan dengan maksud untuk :

1. Menganalisis beberapa faktor yang mempengaruhi proses pengembangan perangkat lunak.
2. Membantu dalam pengumpulan data untuk penelitian saya sebagai salah satu syarat kelulusan program sarjana di Institut Teknologi Sepuluh Nopember, Jurusan Sistem Informasi

Semua data yang terkumpul melalui kuesioner ini adalah untuk tujuan akademis. Semua informasi yang Saudara/i berikan akan saya gunakan dengan sebaik-baiknya untuk penelitian ini. Untuk itu, besar harapan saya agar kuesioner ini dijawab dengan seakurat mungkin sesuai dengan yang Saudara/i alami dan pahami.

Akhir kata, saya ucapkan terima kasih atas partisipasi Anda dalam mengisi kuesioner ini. Semoga hasil penelitian ini dapat bermanfaat bagi kita semua.

Surabaya, Agustus 2015

Son Navy Ramadhan

NRP. 5209 100 029

----- **Demografi Responden** -----

Nama :

.....

Jenis Kelamin : L / P \*

Jabatan :

.....

Nama Proyek

Pengembangan Perangkat Lunak :

.....

\*) *lingkari salah satu*

Anda dimohon untuk memberi tanda silang (X) pada kolom penilaian subjektif untuk tiap-tiap faktor yang telah diuraikan dalam pernyataan berikut.

*Keterangan Nilai:*

1 = Sangat Tidak Setuju, 2 = Tidak Setuju, 3 = Ragu-ragu, 4 = Setuju, 5 = Sangat Setuju

### A. METODE PENGEMBANGAN PERANGKAT LUNAK

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
MPPL	1	Kebutuhan membuat rancangan dalam metode pengembangan perangkat lunak sangat kompleks. (Semakin kompleks metode pengembangan, maka semakin sangat setuju).					
	2	Banyak algoritma pengolahan data yang efektif pada metode pengembangan perangkat lunak yang digunakan. (Semakin efektif metode pengembangan, maka semakin sangat setuju).					
	3	Tim anda merasa familiar (menguasai) dengan metode pengembangan perangkat lunak yang sedang digunakan. (Semakin familiar, maka semakin sangat setuju).					

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
	4	Tim anda paham dengan Pemrograman berorientasi objek atau <i>Object Oriented (OO) programming</i> . (Semakin paham dengan Pemrograman berorientasi objek, maka semakin sangat setuju).					
	5	Tim anda termotivasi dengan metode pengembangan perangkat lunak yang sedang digunakan. (Semakin besar motivasi dalam tim, maka semakin sangat setuju).					
	6	Tim anda sering berselisih pada saat memilih metode pengembangan perangkat lunak. (Semakin tim sering berselisih, maka semakin sangat setuju).					

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
	7	Tim anda mampu menganalisa metode pengembangan perangkat lunak mana yang cocok untuk digunakan. (Semakin besar kapabilitas dan pengetahuan tim, maka semakin sangat setuju).					
	8	Di dalam tim anda terdapat anggota tim yang bekerja paruh waktu. (Semakin banyak anggota tim untuk bekerja paruh waktu, maka semakin sangat setuju).					

## B. TEKNOLOGI PENGEMBANGAN PERANGKAT LUNAK

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
TPPL	1	Kecepatan respon (server) pada aplikasi pengembang perangkat lunak mendukung pekerjaan tim anda. (Semakin pentingnya peningkatan waktu respon, maka semakin sangat setuju).					
	2	Tim anda menggunakan aplikasi pengembang perangkat lunak dengan optimal dan efisien. (Semakin optimal efisiensi dalam penggunaan, maka semakin sangat setuju).					
	3	Tim anda mudah melakukan instalasi aplikasi pengembangan perangkat lunak yang sedang digunakan. (Semakin mudah instalasinya, maka semakin sangat setuju).					

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
	4	Tim anda mudah dalam menggunakan aplikasi pengembangan perangkat lunak yang sedang digunakan. (Semakin mudah penggunaannya, maka semakin sangat setuju).					
	5	Tim anda membutuhkan aplikasi perangkat lunak yang mendukung multi-platform (versi browser, aplikasi mobile, dll atau Windows / OSX / Unix). (Semakin membutuhkan aplikasi multi-platform, maka semakin sangat setuju).					
	6	Tim anda dapat dengan mudah mengubah atau menyesuaikan aplikasi pengembang perangkat lunak sesuai keinginan. (Semakin mudah perubahan atau penyesuaian aplikasi, maka semakin sangat setuju).					

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
	7	<p>Tim anda dapat dengan mudah memberi pengamanan pada aplikasi pengembang perangkat lunak yang sedang digunakan.</p> <p>(contoh, memberi password agar pekerjaan tidak dapat diubah dll.)</p> <p>(Semakin mudah aplikasi diberi pengamanan, maka semakin sangat setuju).</p>					
	8	<p>Tim anda membutuhkan pelatihan untuk menguasai aplikasi pengembang perangkat lunak.</p> <p>(Semakin lama waktu pelatihan yang dibutuhkan, maka semakin sangat setuju).</p>					



Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
	9	Tim anda termotivasi untuk menggunakan teknologi yang dimiliki aplikasi pengembang perangkat lunak yang sedang digunakan. (Semakin besar motivasi dalam tim, maka semakin sangat setuju).					
	10	Tim anda kesulitan dalam memahami bahasa pemrograman yang digunakan pada aplikasi pengembang perangkat lunak. (Semakin sulit bahasa pemrograman, maka semakin sangat setuju).					
	11	Tim anda berpengalaman dalam menggunakan aplikasi pengembang perangkat lunak yang sedang digunakan. (Semakin banyak pengalaman dalam menggunakan aplikasi, maka semakin sangat setuju).					

### C. KOMPONEN PERANGKAT LUNAK

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
KPL	1	<p>Tim anda menggunakan kode yang sudah ada pada saat mengembangkan perangkat lunak.</p> <p>(Semakin sering menggunakan kode yang sudah ada, maka semakin sangat setuju,</p> <p><b>Skala 1:</b> Tidak memerlukan pengulangan kode; <b>Skala 3:</b> menggunakan kode yang sudah ada dengan wajar; <b>Skala 5:</b> kebutuhan yang tinggi dalam menggunakan kode yang sudah ada).</p>					

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
	2	<p>Penggunaan Third-Party code perlu digunakan tim anda dalam mengembangkan perangkat lunak. (contoh, jika mengembangkan aplikasi menggunakan bahasa PHP dengan Framework, Third-Party code-nya adalah framework libraries; jika mengembangkan aplikasi desktop dengan bahasa C#, Third-Party code-nya adalah .Net classes; jika mengembangkan aplikasi menggunakan bahasa Java, Third-Party code-nya adalah JDK dll.) (Semakin sering menggunakan Third-Party Code, maka semakin sangat setuju,</p> <p><b>Skala 1:</b> Tidak diperlukan Third-Party code; <b>Skala 3:</b> Third-Party code digunakan hanya di tim internal; <b>Skala 5:</b> Third-Party code digunakan di tim internal dan pihak ketiga).</p>					

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
	3	Tim anda selalu mengubah kebutuhan saat mengembangkan perangkat lunak. (Semakin sering kebutuhan diubah, maka semakin sangat setuju).					

#### D. SOURCES DARI INTERNET

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
SDI	1	Tim anda menggunakan kode yang beredar di internet pada saat mengembangkan perangkat lunak. (Semakin sering menggunakan kode yang beredar di internet, maka semakin sangat setuju, <b>Skala 1:</b> Tidak memerlukan kode dari internet; <b>Skala 3:</b> menggunakan kode yang beredar di internet dengan wajar; <b>Skala 5:</b> kebutuhan yang tinggi dalam menggunakan kode yang beredar di internet).					

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
	2	<p>Penggunaan Third-Party code yang berasal dari internet perlu digunakan tim anda pada saat mengembangkan perangkat lunak.</p> <p>(Semakin sering menggunakan Third-Party Code yang berasal dari internet, maka semakin sangat setuju,</p> <p><b>Skala 1:</b> Tidak diperlukan Third-Party code; <b>Skala 3:</b> Third-Party code yang berasal dari internet digunakan hanya di dalam tim; <b>Skala 5:</b> Third-Party code yang berasal dari internet digunakan di dalam tim dan pihak ketiga).</p>					

**E. BUDAYA PERUSAHAAN**

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
<b>BP</b>	1	Organisasi anda memiliki tingkat kinerja yang tinggi dan konsisten. (Semakin konsisten tingkat kinerja sebuah organisasi, maka semakin sangat setuju).					
	2	Terdapat komunikasi yang terbuka di dalam organisasi anda. (Semakin terbukanya dalam berkomunikasi, maka semakin sangat setuju).					

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
	3	Anda merasa organisasi memberikan peluang untuk mengasah kemampuan anda (Semakin banyaknya peluang, maka semakin sangat setuju).					
	4	Tanggung jawab yang diberikan organisasi kepada anda sudah tepat sasaran. (Semakin tepat sasaran sebuah tanggung jawab, maka semakin sangat setuju).					
	5	Organisasi anda adil dalam memberikan gaji atau imbalan. (Semakin tinggi keadilan organisasi, maka semakin sangat setuju).					



**F. BUDAYA TIM**

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
BT	1	Terdapat keramahan dan kemudahan dalam bersosialisasi di dalam tim anda. (Semakin tinggi tingkat sosialisasi di dalam tim, maka semakin sangat setuju).					
	2	Sesama anggota di dalam tim anda saling memberikan identitas yang jelas dengan mudah. (Semakin mudahnya menjelaskan identitas anggota, maka semakin sangat setuju).					

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
	3	Tim anda memiliki kepercayaan diri yang tinggi sehingga perasaan “tim paling unggul” muncul. (Semakin tinggi kepercayaan diri di dalam tim, maka semakin sangat setuju).					
	4	Tim anda memiliki sasaran yang tergambar jelas. (Semakin jelas sasaran tim anda, maka semakin sangat setuju).					
	5	Tim anda dapat dengan mudah mengganti anggotanya. (Semakin sering pergantian anggota terjadi, maka semakin sangat setuju).					

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
	6	Stakeholder anda dapat dengan mudah menjalin hubungan yang erat kepada tim anda. (Semakin eratnya hubungan antara stakeholder dan tim, maka semakin sangat setuju).					
	7	Komunikasi antar anggota yang terjadi di dalam tim anda mengalir dengan tepat guna. (Semakin efisien anggota tim berkomunikasi, maka semakin sangat setuju).					
	8	Tim anda dapat dengan mudah menghasilkan ide-ide yang inovatif. (Semakin banyak ide baru yang dihasilkan, maka semakin sangat setuju).					

### G. PENGALAMAN & KEMAMPUAN

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
PDK	1	Anggota tim anda terdiri dari bermacam-macam tipe orang. (Semakin terlihat perbedaan gaya perilaku di dalam tim, maka semakin sangat setuju).					
	2	Programmer di dalam tim anda paham dengan segala jenis pemrograman. (Semakin fasih menguasai berbagai bahasa pemrograman, maka semakin sangat setuju).					

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
	3	Tim anda berpengalaman dengan platform pengembangan perangkat yang sedang digunakan, baik perangkat keras maupun lunak. (Semakin mahir menguasai platform, maka semakin sangat setuju).					
	4	Manajer proyek pada tim anda sering mengawasi pada saat mengembangkan perangkat lunak. (Semakin tinggi kontrol dari manajer, maka semakin sangat setuju).					

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
	5	Manajer proyek pada tim anda merasa familiar (menguasai) dengan aplikasi pengembangan perangkat lunak yang sedang digunakan. (Semakin familiar, maka semakin sangat setuju).					

## H. LINGKUNGAN KERJA

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
LK	1	Lingkungan kerja tim anda mendukung pengembangan perangkat lunak. (Semakin serasi lingkungan kerja anda, maka semakin sangat setuju).					
	2	Di lingkungan kerja, tim anda kesulitan dalam memusatkan perhatian pada proyek yang sedang dikerjakan. (Semakin sulit perhatian terpusat, maka semakin sangat setuju).					

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
	3	Anggota tim sering melakukan “Context Switch” pada saat pengembangan perangkat lunak. (Semakin sering Context Switch dilakukan, maka semakin sangat setuju).					
	4	Tim anda sering melakukan pergantian susunan anggota pada satu lingkungan kerja atau beberapa lingkungan kerja. (Semakin sering anggota tim berganti posisi, maka semakin sangat setuju).					



Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
	5	Lingkungan kerja tim anda mendukung beberapa fasilitas komunikasi seperti tim virtual, atau konferensi video dengan klien dsb. (Semakin lengkap fasilitas komunikasinya, maka semakin sangat setuju).					

# **I. PROYEK PENGEMBANGAN PERANGKAT LUNAK**

<b>Kode</b>	<b>No</b>	<b>Pernyataan</b>	<b>Penilaian Subjektif</b>				
			<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>PPP L</b>	1	Penjadwalan pekerjaan yang terjadi di dalam tim sudah sesuai. (Semakin layak penjadwalan pekerjaan, maka semakin sangat setuju).					
	2	Anggota tim sering mengubah kebutuhan pada saat pengembangan perangkat lunak. (Semakin sering kebutuhan berubah, maka semakin sangat setuju).					

Kode	No	Pernyataan	Penilaian Subjektif				
			1	2	3	4	5
	3	Tim anda memiliki jumlah anggota yang banyak. (Semakin banyak anggota tim anda, maka semakin sangat setuju).					

*Halaman ini sengaja dikosongkan*

## LAMPIRAN B

### Data Responden

MPP L1	MPP L2	MPP L3	MPP L4	MPP L5	MPP L6	MPP L7	MPP L8
4	5	5	4	4	4	5	5
5	5	5	5	5	4	5	5
4	4	3	4	5	3	5	2
5	4	3	5	4	4	4	3
5	3	3	4	3	4	3	4
4	2	5	5	4	1	4	2
5	4	4	5	5	3	5	1
5	5	3	5	5	2	3	5
5	5	4	5	5	2	4	5
5	4	5	4	4	5	4	2
5	2	2	3	4	4	2	5
4	4	5	3	3	1	3	3
4	5	3	5	4	2	4	2
5	5	3	4	4	5	4	1
4	4	4	4	4	2	4	1
4	3	2	4	2	1	2	1
3	2	1	1	3	3	2	1
4	4	4	3	4	2	4	1
4	4	4	4	4	5	4	4
5	5	5	5	5	5	5	5
5	5	5	5	5	4	5	5
5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5
5	4	4	4	4	2	3	2
5	4	5	5	5	5	5	3
5	4	3	4	3	4	4	4
4	4	3	2	4	5	4	4
5	2	4	5	5	2	3	5
5	4	3	3	3	4	3	3

TP PL 1	TP PL 2	TP PL 3	TP PL 4	TP PL 5	TP PL 6	TP PL 7	TP PL 8	TP PL 9	TP PL 10	TP PL 11
4	4	5	5	5	5	5	3	4	4	5
4	5	5	5	4	5	5	3	4	4	5
4	4	5	4	5	5	5	3	4	4	5
5	4	4	4	3	3	4	5	5	4	5
5	4	4	4	4	3	4	5	5	4	5
4	4	5	5	3	5	5	2	4	1	3
5	5	5	5	4	5	5	3	5	1	4
5	5	5	5	4	3	5	1	5	2	2
4	4	2	4	2	2	2	5	4	5	1
3	4	4	4	4	4	3	2	3	2	2
3	4	3	4	5	2	3	2	3	2	4
4	4	4	5	3	4	4	5	4	3	4
5	4	2	3	5	4	5	1	5	2	2
4	4	2	4	1	2	1	5	4	5	1
4	3	4	4	4	4	4	3	4	2	4
3	3	2	2	4	4	3	3	3	3	2
3	3	3	3	3	3	1	1	3	1	1
2	4	4	4	3	4	4	4	2	4	5
2	4	5	5	5	5	5	1	2	3	3
4	5	5	5	4	5	5	5	4	4	5
4	5	5	5	5	5	5	5	4	2	2
4	5	5	5	4	5	5	2	4	1	1
5	5	5	5	5	5	5	4	5	2	4
4	5	5	5	4	5	5	2	4	2	2
4	4	5	4	5	5	5	5	4	4	1
4	5	5	5	4	5	5	5	4	5	1
3	4	4	4	2	2	4	2	3	2	2
3	4	2	4	4	2	2	2	3	2	4
4	5	5	5	5	5	5	5	4	3	5
4	4	4	4	3	3	4	5	4	3	4

KPL1	KPL2	KPL3	SDI1	SDI2
4	5	4	4	4
4	3	4	4	4
5	5	5	5	4
5	4	4	4	4
4	4	4	5	5
5	5	5	5	5
4	5	5	4	4
3	4	4	3	3
4	4	3	4	4
4	4	4	3	4
5	4	4	5	5
4	4	4	4	4
3	3	3	3	2
4	4	4	4	5
3	3	3	2	2
4	5	4	4	4
4	4	4	4	3
2	2	2	1	1
4	4	4	4	3
4	4	3	4	4
4	3	3	3	3
3	3	3	3	3
3	3	3	3	3
3	3	3	4	3
4	4	4	4	5
4	4	4	5	5
4	4	5	4	4
5	4	4	4	4
5	5	4	5	5
5	5	5	4	5

BP1	BP2	BP3	BP4	BP5
4	5	5	4	4
4	4	4	4	3

<b>BP1</b>	<b>BP2</b>	<b>BP3</b>	<b>BP4</b>	<b>BP5</b>
3	2	4	4	5
4	5	5	4	4
4	5	4	5	4
3	4	4	4	4
2	5	4	5	4
2	4	4	3	3
1	4	4	3	3
1	3	4	3	3
2	4	4	5	5
4	4	4	3	3
5	4	5	3	2
4	4	4	4	5
4	4	5	2	2
4	5	4	5	4
4	4	5	3	4
4	4	4	2	2
5	5	4	4	4
4	3	4	4	3
2	4	3	3	4
1	4	4	3	4
2	4	4	3	4
3	5	3	3	4
4	4	3	3	3
4	3	4	4	5
5	3	4	5	4
5	5	5	4	4
2	4	4	5	4
5	4	4	4	4

<b>BT1</b>	<b>BT2</b>	<b>BT3</b>	<b>BT4</b>	<b>BT5</b>	<b>BT6</b>	<b>BT7</b>	<b>BT8</b>
4	4	4	5	5	5	4	5
4	4	3	4	4	3	4	3
5	5	3	5	2	4	5	4



BT1	BT2	BT3	BT4	BT5	BT6	BT7	BT8
5	5	5	5	1	4	4	5
4	2	4	4	2	4	5	5
4	5	4	4	3	5	4	5
5	4	5	4	4	5	4	5
4	4	4	4	4	4	3	4
4	4	5	3	4	4	4	3
4	5	4	3	4	4	4	3
4	4	5	5	3	4	5	5
4	4	4	4	4	5	4	4
2	5	4	2	1	5	2	3
2	4	4	5	3	4	5	5
1	4	4	3	2	2	3	3
1	3	4	5	3	5	4	4
2	4	4	3	2	5	4	3
4	4	4	1	3	4	1	2
3	5	2	4	4	1	4	4
4	4	4	4	4	3	4	4
4	4	3	4	2	2	4	4
5	4	5	3	4	3	3	3
4	4	4	4	2	3	4	4
4	5	5	4	4	3	4	3
4	5	5	4	2	4	4	4
4	4	4	4	3	4	4	4
3	2	4	4	2	5	4	4
4	5	5	4	4	5	4	4
4	5	4	4	4	4	4	5
3	4	4	4	4	4	4	4

PDK1	PDK2	PDK3	PDK4	PDK5
4	4	5	4	3

<b>PDK1</b>	<b>PDK2</b>	<b>PDK3</b>	<b>PDK4</b>	<b>PDK5</b>
5	4	3	3	2
4	5	4	4	4
5	4	5	4	3
4	5	4	4	5
5	5	4	4	4
4	4	4	5	5
4	4	4	3	4
4	3	4	4	4
2	3	3	4	4
4	4	5	5	4
2	4	3	4	4
4	3	2	2	4
2	4	5	5	4
3	3	2	2	4
2	5	4	5	4
4	3	4	3	4
4	2	1	1	4
4	4	3	3	4
4	4	4	3	5
4	3	4	4	4
4	3	4	3	4
4	3	4	4	3
4	4	4	3	4
4	4	3	4	4
4	4	5	5	1
4	4	5	4	3
4	4	5	4	2
4	4	5	5	3
4	4	5	4	2

<b>LK 1</b>	<b>LK 2</b>	<b>LK 3</b>	<b>LK 4</b>	<b>LK 5</b>	<b>PPPL 1</b>	<b>PPPL 2</b>	<b>PPPL 3</b>
4	4	1	4	5	4	2	5

LK 1	LK 2	LK 3	LK 4	LK 5	PPPL 1	PPPL 2	PPPL 3
3	4	3	3	4	3	3	3
5	2	2	4	5	4	2	5
5	4	3	5	5	4	5	4
5	2	3	5	5	5	3	5
5	4	3	4	4	5	2	5
5	4	4	4	4	5	5	5
3	4	4	3	4	4	3	4
3	5	5	3	4	3	4	3
3	5	5	3	4	4	4	4
4	4	2	5	5	5	4	4
3	4	5	4	3	4	5	4
2	1	5	3	2	2	4	2
4	3	4	5	5	4	3	4
2	2	2	3	3	3	3	3
5	3	1	5	4	5	5	5
4	3	2	4	3	4	4	4
2	3	3	1	1	1	4	2
4	2	1	4	3	3	2	4
3	3	3	4	3	4	4	4
4	2	2	4	4	4	4	4
4	5	3	4	4	4	5	4
3	3	2	4	3	3	2	4
4	2	3	4	3	4	3	4
4	2	5	4	3	4	4	3
5	2	4	4	4	4	4	4
4	1	4	5	5	4	4	5
5	1	3	5	4	4	5	4
4	2	4	5	5	5	4	5
5	4	4	4	5	4	4	4

*Halaman ini sengaja dikosongkan*

## LAMPIRAN C

### Hasil Analisis GSCA

Model Fit	
<b>FIT</b>	0.557
<b>AFIT</b>	0.523
<b>GFI</b>	0.976
<b>SRMR</b>	0.216
<b>NPAR</b>	76

### Measurement Model

Variable	Loading			Weight			SMC		
	Estimate	SE	CR	Estimate	SE	CR	Estimate	SE	CR
<b>ProdFactor</b>	<b>AVE = 0.725, Alpha =0.968</b>								
<b>BP4</b>	0.834	0.038	21.91*	0.089	0.021	4.16*	0.695	0.061	11.34*
<b>BP5</b>	0.836	0.058	14.48*	0.067	0.029	2.28*	0.700	0.089	7.89*
<b>BT4</b>	0.877	0.055	15.95*	0.050	0.031	1.65	0.769	0.091	8.42*
<b>BT7</b>	0.857	0.087	9.84*	0.124	0.027	4.6*	0.734	0.130	5.62*
<b>BT8</b>	0.845	0.042	20.01*	0.095	0.026	3.68*	0.714	0.069	10.34*

<b>PDK2</b>	0.794	0.06 3	12.51 *	0.123	0.02 8	4.48*	0.630	0.09 7	6.48*
<b>PDK3</b>	0.857	0.07 1	12.05 *	0.121	0.02 2	5.59*	0.734	0.11 1	6.61*
<b>PDK4</b>	0.875	0.06 0	14.58 *	0.112	0.02 5	4.42*	0.766	0.09 5	8.04*
<b>LK1</b>	0.838	0.05 5	15.32 *	0.092	0.03 4	2.71*	0.703	0.08 6	8.13*
<b>LK4</b>	0.876	0.05 1	17.32 *	0.063	0.02 3	2.77*	0.768	0.08 6	8.93*
<b>LK5</b>	0.853	0.05 5	15.64 *	0.090	0.02 4	3.74*	0.727	0.08 9	8.19*
<b>PPPL1</b>	0.875	0.06 6	13.32 *	0.084	0.02 5	3.31*	0.766	0.10 5	7.26*
<b>PPPL3</b>	0.849	0.06 5	13.02 *	0.067	0.02 6	2.55*	0.720	0.10 4	6.93*
<b>Metode</b>	<b>AVE = 0.509, Alpha =0.828</b>								
<b>MPPL1</b>	0.657	0.16 9	3.88*	0.161	0.03 6	4.52*	0.432	0.18 0	2.4*
<b>MPPL2</b>	0.711	0.10 2	7.0*	0.172	0.02 6	6.6*	0.506	0.13 8	3.66*
<b>MPPL3</b>	0.758	0.10 8	7.02*	0.179	0.03 2	5.57*	0.574	0.15 4	3.72*
<b>MPPL4</b>	0.760	0.10 4	7.29*	0.192	0.03 2	5.91*	0.578	0.14 2	4.07*
<b>MPPL5</b>	0.808	0.04 7	17.11 *	0.192	0.03 0	6.45*	0.652	0.07 6	8.62*
<b>MPPL6</b>	0.517	0.15 6	3.32*	0.132	0.04 4	2.97*	0.268	0.14 6	1.83
<b>MPPL7</b>	0.837	0.05 5	15.2*	0.210	0.03 0	6.89*	0.700	0.08 6	8.15*

<b>MPPL8</b>	0.604	0.15 0	4.01*	0.151	0.03 6	4.24*	0.364	0.16 3	2.24*
<b>Teknologi</b>	<b>AVE = 0.416, Alpha =0.759</b>								
<b>TPPL1</b>	0.392	0.16 0	2.46*	0.094	0.03 5	2.65*	0.154	0.12 4	1.25
<b>TPPL2</b>	0.763	0.08 2	9.26*	0.182	0.02 1	8.86*	0.582	0.12 1	4.81*
<b>TPPL3</b>	0.901	0.04 1	22.24*	0.214	0.03 7	5.82*	0.812	0.07 2	11.3*
<b>TPPL4</b>	0.796	0.06 5	12.22*	0.193	0.02 6	7.36*	0.634	0.10 2	6.24*
<b>TPPL5</b>	0.593	0.14 2	4.16*	0.141	0.03 5	4.07*	0.352	0.15 7	2.24*
<b>TPPL6</b>	0.812	0.06 4	12.69*	0.197	0.02 4	8.34*	0.659	0.10 3	6.42*
<b>TPPL7</b>	0.909	0.03 7	24.57*	0.221	0.03 5	6.32*	0.827	0.06 4	12.86*
<b>TPPL8</b>	0.133	0.25 1	0.53	0.031	0.05 6	0.55	0.018	0.09 5	0.19
<b>TPPL10</b>	-0.063	0.28 4	0.22	-0.013	0.06 5	0.2	0.004	0.08 6	0.05
<b>TPPL11</b>	0.345	0.19 5	1.77	0.082	0.04 4	1.88	0.119	0.13 1	0.91
<b>Komponen</b>	<b>AVE = 0.836, Alpha =0.902</b>								
<b>KPL1</b>	-0.890	0.04 5	20.0*	-0.346	0.02 1	16.16*	0.793	0.07 7	10.31*
<b>KPL2</b>	-0.933	0.02 4	38.39*	-0.382	0.04 1	9.41*	0.870	0.04 5	19.39*
<b>KPL3</b>	-0.919	0.03 0	30.96*	-0.365	0.02 9	12.4*	0.845	0.05 3	15.94*

<b>Internet</b>	<b>AVE = 0.921, Alpha =0.911</b>								
<b>SDI1</b>	0.963	0.02 0	48.89 *	0.540	0.04 1	13.22 *	0.927	0.03 7	24.88 *
<b>SDI2</b>	0.957	0.02 4	40.61 *	0.502	0.04 1	12.14 *	0.916	0.04 4	20.84 *

CR\* = significant at .05 level

### Structural Model

<b>Path Coefficients</b>			
	<b>Estimate</b>	<b>SE</b>	<b>CR</b>
<b>Metode-&gt;ProdFactor</b>	0.143	0.234	0.61
<b>Teknologi-&gt;ProdFactor</b>	-0.063	0.224	0.28
<b>Komponen-&gt;ProdFactor</b>	-0.401	0.187	2.15*
<b>Internet-&gt;ProdFactor</b>	0.558	0.178	3.13*

CR\* = significant at .05 level

### R square of Latent Variable

<b>ProdFactor</b>	0.808
<b>Metode</b>	0
<b>Teknologi</b>	0
<b>Komponen</b>	0
<b>Internet</b>	0

### Means Scores of Latent Variables



<b>ProdFactor</b>	3.846
<b>Metode</b>	4.060
<b>Teknologi</b>	4.073
<b>Komponen</b>	3.910
<b>Internet</b>	3.818

<b>Correlations of Latent Variables (SE)</b>					
	<b>ProdFactor</b>	<b>Metode</b>	<b>Teknologi</b>	<b>Komponen</b>	<b>Internet</b>
<b>ProdFactor</b>	1	-0.051 (0.186)	0.020 (0.178)	-0.820 (0.058)*	0.875 (0.065)*
<b>Metode</b>	-0.051 (0.186)	1	0.679 (0.123)*	0.290 (0.168)	-0.064 (0.191)
<b>Teknologi</b>	0.020 (0.178)	0.679 (0.123)*	1	0.069 (0.144)	0.024 (0.159)
<b>Komponen</b>	-0.820 (0.058)*	0.290 (0.168)	0.069 (0.144)	1	-0.817 (0.055)*
<b>Internet</b>	0.875 (0.065)*	-0.064 (0.191)	0.024 (0.159)	-0.817 (0.055)*	1

\* significant at .05 level

## **BAB VII**

### **KESIMPULAN DAN SARAN**

Bab ini berisi mengenai simpulan yang didapatkan dari hasil penelitian yang telah dilakukan. Simpulan ini diharapkan dapat menjawab tujuan yang telah ditetapkan di awal penelitian. Saran diberikan untuk digunakan dalam penelitian selanjutnya

#### **7.1 Kesimpulan**

Dari pelaksanaan penelitian tugas akhir ini didapatkan kesimpulan:

1. Internet pada pengembangan perangkat lunak memiliki pengaruh positif yang kuat terhadap *productivity factor*. Terlihat dari hasil analisis *path coefficients* pada aplikasi GeSCA keterkaitan nilai Estimate dan CR antar kedua variabel tersebut bernilai 0,558 dan 3,13\*, nilai tersebut menyebabkan Internet berdampak langsung pada meningkatnya nilai *Productivity Factor*. Hal ini mengindikasikan apabila sebuah tim *software developer* menggunakan *tools* atau *code* yang didapat dari Internet, maka nilai *Productivity Factor* akan meningkat.
2. Komponen Pengembangan Perangkat Lunak memiliki pengaruh negatif yang kuat terhadap *Productivity Factor*. Terlihat dari hasil analisis *path coefficients* pada aplikasi GeSCA keterkaitan nilai Estimate dan CR antar kedua variabel tersebut bernilai -0,401 dan 2,15\*, nilai tersebut menyebabkan. Komponen Perangkat Lunak akan berdampak langsung pada turunnya nilai *Productivity Factor*. Hal ini mengindikasikan apabila sebuah tim *software developer* menggunakan Komponen Perangkat Lunak dengan efektif, maka nilai *Productivity Factor* akan menurun.

3. Pengaruh Teknologi pengembangan perangkat lunak terhadap *Productivity Factor* adalah positif, sedangkan pengaruh Metode pengembangan terhadap *Productivity Factor* adalah negatif. Terlihat dari hasil analisis *path coefficients* pada aplikasi GeSCA keterkaitan nilai Estimate kedua variabel tersebut terhadap *Productivity Factor* masing-masing bernilai -0,063 dan 0,143. Namun, kedua variabel tersebut memiliki pengaruh yang lemah terhadap variabel *Productivity Factor*, karena hasil analisis *path coefficients* menunjukkan CR Teknologi pengembangan bernilai 0,28 dan Metode pengembangan bernilai 0,61. Dengan kata lain, semakin tinggi penggunaan Teknologi Pengembangan Perangkat Lunak, maka semakin rendah nilai *Productivity Factor* dan semakin tinggi penggunaan Metode Pengembangan Perangkat Lunak, maka semakin tinggi pula nilai *Productivity Factor*, dengan pengaruh yang lemah.
4. Berdasarkan hasil analisis statistika inferensial di atas, dapat disimpulkan faktor-faktor yang menyebabkan turunnya nilai *effort rate* atau *productivity factor* adalah penggunaan Metode Pengembangan Perangkat Lunak dan Komponen Perangkat Lunak. Sedangkan, faktor-faktor yang menyebabkan naiknya nilai *effort rate* atau *productivity factor* adalah penggunaan Teknologi Pengembangan Perangkat Lunak dan *tools* atau *code* yang berasal dari Internet.

## 7.2 Saran

Dari pelaksanaan penelitian tugas akhir ini dapat diberikan saran untuk penelitian selanjutnya antara lain:

1. Pada penelitian ini penulis menerapkan variabel laten yang digunakan pada konseptual model yang mengadopsi dari jurnal penelitian sebelumnya [9]. Hasil dari penelitian ini sudah cukup menggambarkan keterkaitan antar variabel MPPL, TPPL, KPL, SDI

dengan *productivity factor* sebesar 52,3 %, akan tetapi masih ada sekitar 47,7 % dari variabel lain yang bisa mempengaruhi model dalam penelitian Tugas Akhir ini. Penulis menyarankan untuk dilakukannya penelitian lanjutan pada tiap-tiap faktor yang lebih detail.

2. Penulis juga menyarankan kepada *software developer* untuk mempertajam fungsional dari Komponen Perangkat Lunak dan memilih Metode Pengembangan yang tepat agar waktu dan tenaga yang dibutuhkan dalam *software development* dapat diminimalisir.

*Halaman ini sengaja dikosongkan*

## DAFTAR PUSTAKA

- [1] C. L. D. S. d. C. A. M. Barbosa da Silva, "Applying The Use Case Points Effort Estimation Technique To Avionics Systems," *IEEE*, 2008.
- [2] S. Tockey, *Return of Software: Maximizing the Return on Your Software Investment*, Prentice Hall, 2004.
- [3] E. Muhardin, *Estimasi Proyek Software*, 2011.
- [4] B. Anda, "Proceedings of Empirical Assessment in Software Engineering," *Comparing effort estimates based on use cases with expert estimates*, p. 13, 2002.
- [5] G. Karner, "Objective System SF AB," *Resource Estimation for Objectory Projects*, 1993.
- [6] G. a. W. J. Schneider, *Applying Use Cases – A Practical Guide*, Addison-Wesley, 1998.
- [7] R. K. Clemmons, *Project Estimation With Use Case Point*, 2006: Diversified Technical Services, Inc.
- [8] M. Ochodek, J. Nawrocki and K. Kwarciak, "Simplifying Effort Estimation Based on Use Case Points," *Sciencedirect*, 2011.
- [9] A. P. Subriadi, Sholiq and P. A. Ningrum, "Critical Review of The Effort Rate Value in Use Case Point Method for Estimating Software Development Effort," *JATIT*, 2013.
- [10] D. a. M. W. E. Galorath, *Software Sizing, Estimation and Risk Management*, Auerbach, 2006.
- [11] S. M. F. I. K. Kusumoto, "Estimating Effort by Use Case Points: Method, Tool and Case Study," *IEEE*, 2006.
- [12] E. R. Carroll, "Estimating Software Based on Use Case

- Points," in *Object-Oriented, Programming, Systems, Languages, and Object Oriented Programming Systems Languages and Applications (OOPSLA) Conference*, San Diego, CA, 2005.
- [13] Y. A. M. K. M. Yavari, "Software Complexity Level Determination Using Software Effort Estimasi Use Case Points Metrics," *IEEE*, 2011.
  - [14] S. Nageswaran, "Test Effort Estimation," [www.cognizant.com/cogcommunity/presentations/Test\\_Effort\\_Estimasi.pdf](http://www.cognizant.com/cogcommunity/presentations/Test_Effort_Estimasi.pdf), 2001.
  - [15] J. Sarwono, *Metode Penelitian Kuantitatif dan Kualitatif*, Yogyakarta: Graha Ilmu., 2006.
  - [16] J. Sulistyio, 6 Hari Jago SPSS 17, Yogyakarta: Cakrawala, 2010.
  - [17] R. L. & S. M. T. Holbert, *Structural equation modeling in the communication sciences, 1995-2000.*, New York: Human Communication Research, 2002.
  - [18] J. a. A. R. Hair, *Multivariate Data Analysis*, vol. Vol.9, Eaglewood Cliffs, NJ: Prentice-Hall, 1998, pp. 62-81.
  - [19] H. Wold, Partial least squares. In: Kotz, S., Johnson, N.L. (Eds.), *Wiley, New York: Encyclopedia of Statistical Sciences.*, 1985.
  - [20] I. Ghozali and K. Aprilia, *Generalized Structured Component Analysis (GeSCA), Model Persamaan Struktural Berbasis Komponen.*, Semarang: Badan Penerbit - Undip, 2013.
  - [21] J. Cohen, K. Dunbar and J. McClelland, "Onthecontrolofautomaticprocesses: aparalldistributed processing account of the Strooeffect," *Psychol. Rev.*, vol. 97, p. 332–61, Maret 1990.
  - [22] L. a. B. P. Hu, "Cutoff Criteria for Fit Indexes in Covariance Structure Analysis: Conventional

- Criteria Versus New Alternatives," *Structural Equation Modeling*, vol. 6, no. 1, pp. 1-55, 2 Februari 1999.
- [23] Amirullah, *Metodologi Penelitian Manajemen: Disertai Contoh Judul Penelitian dan Proposal*, 1st ed., Malang: Bayumedia Publishing, 2013.
  - [24] Agilis Solutions, *Software Estimating Based on Use Case*, vol. III, no. 12, 2009.
  - [25] S. F. a. G. Engels, Revised Use Case Point Method-Effort Estimation in Development Projects for Business Application, 2nd ed., Proceedings of the CONQUEST, 2008.
  - [26] F. D. M. S. A. R. S. a. H. S. Marimin, "Added Value and Performance Analyses of Edamame Soybean Supply Chain: A Case Study," *Operations and Supply Chain Management*, vol. 3, no. 3, pp. 148-162, 2010.
  - [27] S. Wagner dan M. Ruhe, "A Systematic Review of Productivity Factors in Software Development," *Productivity Factors*, 2008.
  - [28] L. d. D. P. Gay, *Research Methods for Business and Management*, New York: MacMillan Publishing Company, 1992.
  - [29] W. Sebastian, "A Synoptic View" International Journal of Finance and Economics. "Thirty Years of Current Account Imbalances.," *Globalization*, 2004.



*Halaman ini sengaja dikosongkan*

## BIODATA PENULIS



Son Navy Ramadhan Kurnia Al Akbar yang akrab dipanggil Sonna dilahirkan di Surabaya pada tanggal 18 Maret 1991, merupakan anak pertama dari dua bersaudara. Penulis telah menempuh pendidikan formal yaitu di TK Al-Muttaqien Surabaya, SD Al-Muttaqien Surabaya, SMP Negeri 6 Surabaya dan SMA Muhammadiyah 2 Surabaya. Selepas lulus SMA, penulis melanjutkan pendidikan jenjang S1 dengan mengikuti PMDK Mandiri ITS dan diterima di Jurusan Sistem Informasi ITS Surabaya pada tahun 2009 dan terdaftar dengan NRP 5209-100-029.

Selama menempuh perkuliahan, penulis aktif dalam organisasi Himpunan Mahasiswa Sistem Informasi (HMSI) yaitu mendapat amanah sebagai anggota Departemen Kewirausahaan (2010,2011), dalam organisasi Badan Eksekutif Mahasiswa Fakultas Teknologi Informasi (BEM FTIF) yaitu mendapat amanah sebagai anggota Departemen Pengembangan Riset dan Teknologi (2011,2012) dan dalam unit kegiatan mahasiswa Workshop of Entrepreneurship and Technology (WE&T) (2010,2011). Sebagai penelitian tugas akhir di Jurusan Sistem Informasi, penulis memilih bidang minat Manajemen Sistem Informasi.

Jika ada pertanyaan mengenai tugas akhir ini, penulis dapat dihubungi melalui email [sonnavyrkaa@gmail.com](mailto:sonnavyrkaa@gmail.com).